

**O'ZBEKISTON RESPUBLIKASI OLIY VA O'RTA MAXSUS TA'LIM
VAZIRLIGI
BUXORO DAVLAT UNVERSITETI**

**“MA'LUMOTLAR TUZILMASI” FANIDAN
O'QUV USLUBIY QO'LLANMA**



Buxoro2019

Tuzuvchilar:

BuxDU Fizika-matematika fakulteti “Axborot texnologiyalari” kafedrası o`qituvchisi H.I.Eshanqulov

BuxDU Fizika matematika fakulteti Kasptalimi infarat va axborot texnologiyalari yo`nalishi 1-1PMIK guruh talabasi U.M.Kadirova

Taqrizchilar:

BuxDU Fizika-matematika fakulteti “Axborot texnologiyalari” kafedrası mudiri O.I.Jalolov

B.M.T.I “Axborot kommunikatsiya texnologiyalari” kafedrası katta o`qituvchisi T.F.Sohibov

1-MA'RUZA:MA'LUMOTLAR TUZILMALARI VA ALGORITMLARI: ASOSIY TUSHUNCHALAR VA TA'RIFLAR.....	5
OPERATIV XOTIRADAGI MT	8
2-MA'RUZA:MA'LUMOTLARNING CHIZIQLI TUZILMALARI. MA'LUMOTLARNING YOZUV TURLARI.JADVALLAR.	10
3-MA'RUZA:YARIM STATIK MA'LUMOTLAR TUZILMALARI.....	15
4-MA'RUZA.GRAFLAR VA DARAXTLAR.....	21
5-MA'RUZA:KO'P BOG'LIQ RO'YXAT BILAN AKS ETTIRILADIGAN MA'LUMOTLAR TUZILMALARI.	34
6-MA'RUZA:MA'LUMOTLARNI TARTIBGA SOLISH TUSHUNCHASI. MA'LUMOTLAR TUZILMASINI SARALASH USULLARI.	38
7-MA'RUZA:SARALASH USULLARINI TANLASHDA HISOBGA OLINADIGAN OMILLAR.....	47
8-MA'RUZA:MA'LUMOTLARNI IZLASH ALGORITMLARI.KETMA-KET IZLASH. IZLASHNING TEZLASHTIRILGAN USULLARI.....	53
9-MA'RUZA:MA'LUMOTLARDAN BEVOSITA ERKIN FOYDALANADIGAN IZLASH USULI.....	60
1-LABARATORIYA MASHG'ULOTI:MA'LUMOTLARNING ODDIY SOZLANGAN TOIFALARI	66
2-LABARATORIYA MASHG'ULOTI:MANTIQUIY MA'LUMOTLAR TURLARI. MASSIV ELEMENTLARI.....	67
3-LABARATORIYA MASHG'ULOTI:YOZUVLAR USTIDA BAJARILADIGAN AMALLAR.	69
4-LABARATORIYA MASHG'ULOTI:MA'LUMOTLARNING STEK TURLARI.....	72
5-LABARATORIYA MASHG'ULOTI:MA'LUMOTLARNING STEK TURLARI.....	73
6-LABARATORIYA MASHG'ULOTI:MA'LUMOTLARNING NAVBAT TURLARI.....	75
7-LABARATORIYA MASHG'ULOTI: DINAMIK MA'LUMOTLAR TUZILMASI. BIR BOG'LAMLI RO'YHAT BOSHIDAN ELEMENTNI O'CHIRISH.	76

8-LABARATORIYA MASHG'ULOTI: BINAR DARAXTLARNI QURISH. BINAR DARAXT YARATISH FUNKSIYASI.....	77
9-LABARATORIYA MASHG'ULOTI: MA'LUMOTLARNI DARAXTSIMON TAQDIM ETISHDA	78
10-LABARATORIYA MASHG'ULOTI:TASHQI SARALASH	80
11-LABARATORIYA MASHG'ULOTI:SARALASH USULLARINI TANLASHDA HISOBGA OLINADIGAN OMILLAR. YOZUVNI JOYLASHTIRISH USULLARI	81
12-LABARATORIYA MASHG'ULOTI:KETMA-KET IZLASH.....	82
13-LABARATORIYA MASHG'ULOT:IKKILANGAN DARAXT BO'YICHA IZLASH.	84
14-LABARATORIYA MASHG'ULOTI:MA'LUMOTLARDAN BEVOSITA ERKIN FOYDALANADIGAN IZLASH USULI	85
GLOSSARIY.....	86
FOYDALANILADIGAN ADABIYOTLAR RO'YXATI	99

1-Ma'ruza:Ma'lumotlar tuzilmalari va algoritmlari: asosiy tushunchalar va ta'riflar.

Reja:

1. “Ma'lumotlar tuzilmasi ” fanining predmeti.
2. Ma'lumotlar tuzilmasini EHM xotirasida ifodalashning asosiy tushuncha va ta'riflari..
3. Ma'lumotlarni tuzilishi jihatidan sinflari.
4. Ma'lumotlar tuzilmasi ustida bajariladi amallar.

Tayanch iboralar: *AKT, Ma'lumotlar tuzilmalari, kommunikatsiya*

O'quv maqsadi:

Ta'limiy: Talabalarda “Ma'lumotlar tuzilmasi ” fanining mohiyati, maqsadi va vazifalari hamda zamonaviy axborot texnologiyalarining hozirgi zamondagi o'rni va rivojlanish istiqbollari, ma'lumotlarni ifodalash bosqichlari. Ma'lumotlar tuzilmasini klassifikatsiya qilish va ular ustida amallar to'g'risida bilim va ko'nikmalarni shakllantirish.

Tarbiyaviy: Ma'lumotlar tuzilmasi orqali bajariladigan ishlar, ularning ahamiyatini tushuntirish orqali talabalarni bu fanga qiziqish ruhida tarbiyalash;.

Rivojlantiruvchi: Talabalarning “Ma'lumotlar tuzilmasi ” fanidan olgan bilimlarini mutaxassislik sohasidagi ishlarida qo'llash orqali rivojlantirish.

1. “Ma'lumotlar tuzilmasi ” fanining predmeti. O'quv fanining maqsadi va vazifalari

Talabalarda ma'lumotlar tuzilmasini tashkil etish tamoyillari, ularning EHM xotirasida ifodalanish tartibi, hamda ma'lumot turlari, ularning qiymat qabul qilish chegarasi, ularga EHM xotirasida ajratiladigan joy va ularga murojaat qilish tartibi, ma'lumotlar qidirish va saralash algoritmlarining samarador usullari bo'yicha ko'nikma va malakalarni shakllantirishdan iborat.

Fanning vazifasi – talabalarga dasturlash jarayonida turli turdagi

ma'lumotlardan foydalanishni, ularni yaratish va murojaat qilish bilishi, ro'yxat, daraxt tuzilmasida ma'lumotlarda qidirish, saralash algoritmlarni qo'llashni o'rganishdan iborat.

Fan bo'yicha talabalarning bilimiga, ko'nikma va malakasiga qo'yiladigan talablar: Oliy o'quv yurtlarining tahsil olayotgan talabalarida: ma'lumot tuzilmasida element, uning joylashgan o'rni, elementga murojaat qilish tartibi, ma'lumotlar tuzilmasini xotirada yaratish, uni xotiradan yo'qotish (xotirani bo'shatish), ma'lumotlar tuzilmasidani kerakli ma'lumotni qidirish usullari va saralash algoritmlaridan foydalanish bo'yicha tasavvurga ega bo'lishi;

Rekursiv ro'yxat tushunchalari bilan tanishish, ro'yxatni hosil qilish, ro'yxatda elementlarida qidiruv usullarini qo'llash, ro'yxat elementlari orasidagi munosabatlarni bilishi. Ro'yxat elementlarini bo'yicha optimal qidirish va saralash algoritmlarini qo'llash bo'yicha ko'nikma va tajribaga ega bo'lishi kerak

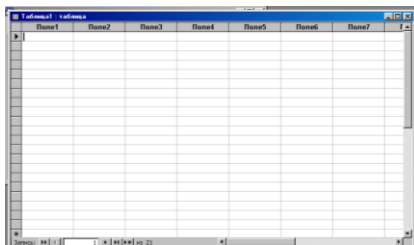
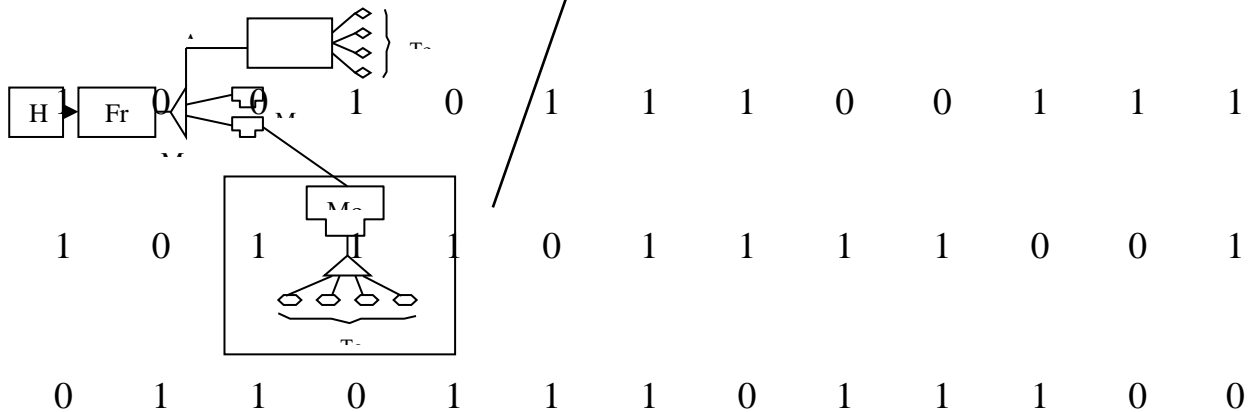
Fanning o'quv rejadagi boshqa fanlar bilan o'zaro bog'liqligi va uslubiy jihatdan uzviy ketma-ketligi: "Ma'lumotlar tuzilmasi " fani 3-semestrlarda o'qitiladi. Talabalarni yuqorida keltirilgan tasavvur, bilim va ko'nikmalarga erishish uchun nazariy, amaliy, tajriba mashg'ulotlari ko'zda tutiladi. "Ma'lumotlar tuzilmasi " fanini o'rganishda matematika va tabiiy (oliy matematika, informatika va axborot texnologiyalari, dasturlash texnologiyasi, dasturlash asoslari) kabi fanlardan olingan bilim va ko'nikmalarga ega bo'lishi talab etiladi.

Fanni o'qitishda zamonaviy axborot va pedagogik texnologiyalar: Talabalarning "Ma'lumotlar tuzilmasi " fanini o'zlashtirishlari uchun o'qitishning ilg'or va zamonaviy usullaridan foydalanishni, yangi informatsion-pedagogik texnologiyalarni tadbiq qilish muxim ahamiyatga egadir. Fanni o'zlashtirish darslik, o'quv va uslubiy qo'llanmalar ma'ruza matnlari, tarqatma materiallar, virtual stendlar hamda yuqori saviyali dasturlash tillaridan foydalaniladi. Ma'ruzada ilg'or pedagogik texnologiyalardan foydalaniladi.

2. Ma'lumotlar tuzilmasini EHM xotirasida ifodalashning asosiy tushuncha va ta'riflari.



Tarkibi va murakkabligidan bo'liq bo'lmagan holda har qanday ma'lumot EHM xotirasiga ikkilik razryadlar yoki bitlar ketma-ketligida ifodalanadi, ularning qiymatlari mos ravishda ikkilik sonlar bo'ladi.



Bitlar ketma-ketligi ko'rinishida qaralayotgan ma'lumotlar juda sodda tashkil etilgan yoki boshqacha aytganda oddiy strukturalashtirilgan. Murakkab

ma'lumotlarni bitlar ketma-ketligida ifodalash va tekshirish inson uchun noqulay tushuncha sanaladi.

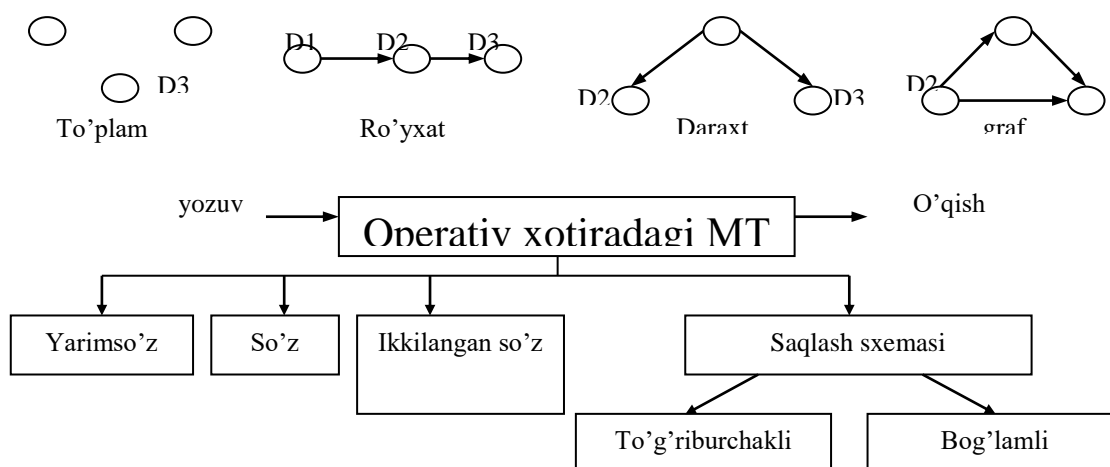
Ma'lumotlar tuzilmasining bunday ta'rifi mumkin bo'lgan barcha yondoshuvlarni qamrab oladi, biroq har bir aniq masalada u yoki bu aspektdan foydalaniladi.

Umumiy holda ma'lumotlar tuzilmasining mantiqiy va mos ravish uning fizik strukturasi orasida farq mavjud, qaysiki ular orasida farqning darajasi tuzilmaning o'zidan va u akslantiriladigan muhitdan bog'liq. Bu farq mantiqiy strukturani fizik va aksincha fizik strukturani mantiqiy akslantirishni amalga oshirishni tekshiruvchi protseduralarda mavjud. Bu funktsiyalar akslantirishdan tashqari fizik strukturaga ruxsat va ular ustida turli amallarni bajarish ta'minlaydi, ya'ni har bir amalda ma'lumotlarning fizik yoki mantiqiy tuzilmasiga qo'llanilishi qaraladi. *

3. Ma'lumotlarni tuzilishi jihatidan qanday sinflarga ajratamiz.

MT klassifikatsiya qilishda asosiy belgi bu ma'lumotlar tuzilmasini dastur ishlashi mobaynida o'zgarishi hisoblanadi. Masalan, agar dastur bajarilishi mobaynida elementlar soni va/yoki ular orasidagi munosabatlar o'zgarsa, u holda bunday MT dinamik ma'lumotlar tuzilmasi, aks holda statistik ma'lumotlar tuzilmasi deyiladi.

MTga misollar:



Operativ xotira massivni ifodalaydi.

So'z – bir vaqtning o'zida qayta ishlanishi mumkin bo'lgan minimal sondagi

bitdir.

4. Ma'lumotlar tuzilmasi ustida qanday amallar bajariladi.

Ma'lumotlar tuzilmasini yaratishni misollar orqali tahlil qilamiz. Masalan PL/1 dagi DECLARE N FIXED DECIMAL operatori N o'zgaruvchi uchun adres sohani ajratishga olib keladi. FORTRAN da (Integer I), PASCAL da (I:integer), C da (int I) tipni bunday yozish natijasida mos o'zgaruvchi uchun xotiradan joy ajratiladi. Dasturda ma'lumotlar tuzilmasini yangilash (qayta yangilash) uchun tizimli dasturlash vositalari yoki kompilyatsiya bosqichida, yoki qaysiki mos o'zgaruvchi yangilanadigan protsedura blokini faollashtirishda avtomatik tarzda xotira ajratiladi. Dasturchining o'zi dasturlash tizimining xotirani ajratish/bo'shatish protsedura/funksiyalaridan foydalanib ma'lumotlar tuzilmasi uchun xotiradan joy ajratishi mumkin. Ob'ektga yo'naltirilgan dasturlash tillarida yangi ob'ekt yaratishda uning uchun yaratish va yo'qotish protsedurasi aniqlangan bo'lishi kerak.

Xulosa iboratki qaysi dasturlash tilidan foydalanishdan qat'iy nazar dasturdagi mavjud ma'lumotlar tuzilmasi "hech narsada" ko'rinmaydi, uning ko'rinishi uchun tuzilma ochiq yoki yopiq holda tuzilma yaratish operatorlari orqali e'lon qilinadi. Buning natijasida dasturdagi tuzilmaning barcha nusxalarini joylashtirish uchun xotira ajratiladi.

YUqorida ko'rsatilgan to'rta amal barcha tip va strukturalar uchun zaruriy sanaladi. Maxsus amallar qaralayotgan aniq strukturalarga alohida qaraladi.

Nazorat savollari:

1. Ma'lumotlar tuzilmasi deganda nimani tushunasiz?
2. Ma'lumotlarni tasvirlash bosqichlarini keltirib o'ting.
3. Ma'lumotlar tuzilmasi klassifikatsiyasi qanday amalga oshiriladi?
4. Ma'lumotlar tuzilmasini foydalanuvchi dasturidagi klassifikatsiyasi.
5. Ma'lumotlar tuzilmasini operativ xotiradagi klassifikatsiyasi.
6. Ma'lumotlar tuzilmasini tashqi xotiradagi klassifikatsiyasi.
7. Qanday ma'lumotlar dinamik yoki statik turdagi ma'lumotlar tuzilmasi deyiladi?
8. Ma'lumotlar tuzilmasining muhim belgilarini izohlang?

9. Qanday ma'lumotlar bog'langan yoki bog'lanmagan turdagi ma'lumotlar tuzilmasi deyiladi?
10. Ma'lumotlarning fizik yoki mantiqiy tuzilmasi nima?

2-Ma'ruza: Ma'lumotlarning chiziqli tuzilmalari. Ma'lumotlarning yozuv turlari. Jadvallar.

Reja:

1. Vektorlar.
2. Massiv.
3. Yozuv.
4. Jadval.

O'quv maqsadi:

Ta'limiy: Talabalarga vektor, massiv, yozuv va jadval ma'lumot turlari haqida ma'lumot berish.

Tarbiyaviy: Talabalarni vatanparvarlik ruhida tarbiyalash.

Rivojlantiruvchi: Talabalarning aqliy faoliyatini rivojlantirish.

1. Vektorlar

Vektor – bu eng sodda statik va chiziqli tartiblangan tuzilmadir. Bu tuzilmadagi elementlar orasidagi munosabat ularning qat'iy ketma-ketlik ko'rinishida ifodalanishidir (qarang, chizma).

N	N+1	N+2	...	N+M
---	-----	-----	-----	-----

Vektorning har bir elementi uning vektordagi o'rnini aniqlovchi (ko'rsatuvchi) o'zining indeksiga ega bo'ladi. Indekslar butun son bo'lganligi tufayli ular ustida bir necha amallarni amalga oshirish mumkin hamda murojaatni mantiqiy bosqichida elementni tuzilmadagi o'rnini aniqlash mumkin. Vektorning elementiga murojaat qilish uchun vektorni nomini va elementning indeksini ko'rsatish kifoya bo'ladi.

Dasturda vektorni e'lon qilish uchun uning nomini, elementlar sonini va ularning turini ko'rsatish lozim

Masalan:

var

M1: Array [1..100] of integer;

M2: Array [1..10] of real;

Vektor elementlarning barchasi faqatgina bitta turga tegishli ma'lumotlardan iborat hamda ularning soni oldindan aniq bo'lishi lozim.

2. Massivlar

Umuman olganda massiv elementi bu vektor elementi bo'lib, uning elementi ham tuzilma elementi bo'lib hisoblanadi.

B_{11}	B_{12}	...	B_{1N}
B_{21}	B_{22}	...	B_{2N}
B_{31}	B_{32}	...	B_{3N}
B_{41}	B_{42}	...	B_{4N}

Ikki o'lchamli massiv elementiga murojaatni amalga oshirish uchun uning indeksi qiymatlarini zarur bo'ladi. Fizik bosqichda ikki o'lchamli massiv ham xuddi bir o'lchamli (vektor) massiv kabi ko'rinishga ega bo'ladi hamda translyatorlar massivni qator yoki ustun ko'rinishida ifodalaydi.

3. Yozuv

Yozuv – maydon deb ataluvchi chekli sondagi ma'lumotlar tuzilmasidir. Yozuv ketma-ket turdagi ma'lumotlar tuzilmasini ifodalab, mantiqiy tasvirlanishda ham fizik tasvirlanishda ham tuzilma elementlari ketma-ket joylashgan bo'ladi. YOzuvning massivdan farqi shundan iboratki, uning elementlari turli turlarga tegishli bo'lgan to'plamdan iborat bo'lishi mumkin. Yozuvda ma'lumot elementlarini ko'pincha yozuv maydonlari deb xam ataladi.

Yozuvni e'lon qilish mantiqiy ko'rinishi quyidagicha:

<tur nomi> = record <maydonlar ro'yxati> end

maydonlar orasiga ; belgisi qo'yiladi.

Masalan:

Type

BirthDay = Record

Day, Month:Byte;

```
Year:Word;  
End;  
Var  
a,b:BirthDay;  
Begin  
a.day:=27;  
b.year:=1939;  
End.
```

Maydonlarga murojaatni soddalashtirish uchun with operatoridan foydalaniladi.

Masalan:

```
With a Do  
Begin  
Day:=27;  
Month:=4;  
Year:=1985;  
End.
```

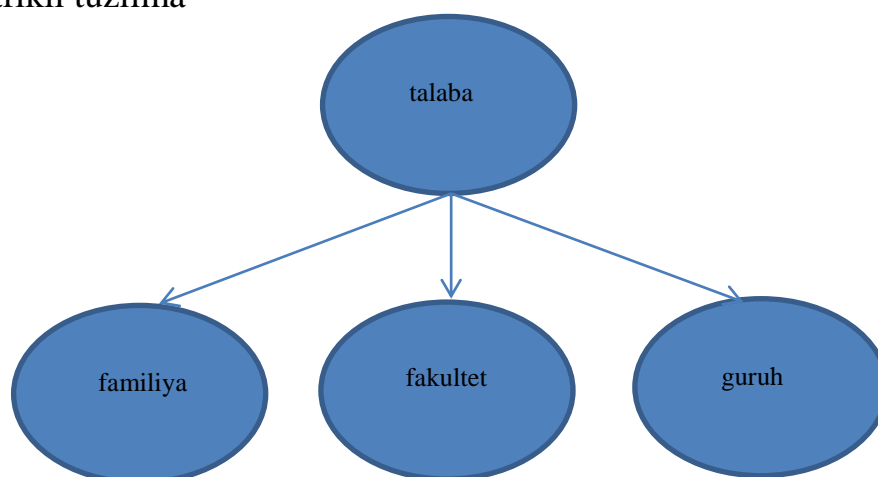
Quyidagi masalani tahlil qilib chiqaylik:

YOZuvning mantiqiy tuzilmasini grafik ko'rinishda xam jadval ko'rinishida xam ifodalash mumkin, ya'ni

Mantiqiy tuzilma

Nomer	Familiya	Fakultet	Guruh
-------	----------	----------	-------

Grafikli tuzilma



Yozuv elementlarini o'zi xam yozuvdan iborat bo'lishi mumkin. Bu holatda murakkab ierarxik ma'lumotlar tuzilmasi vujudga keladi.

Talaba haqida quyidagi ma'lumotlarni o'z ichiga oluvchi yozuvni to'ldirish talab qilingan bo'lsin: N – talaba tartib raqami; talaba Ismi, bu erda talab familiyasi, ismi, otasining ismi bo'lsin; talabning anketa ma'lumotlari, ya'ni tug'ilgan yili, tug'ilgan joyi, ota-onasi: onasi, otasi; Fakul'teti; Guruhi; fanlardan sessiyada olgan baholari, masalan chet tili, informatika, matematika va boshq.

Yuqoridagi ma'lumotni ifodalash natijasida to'rt bosqichli ierarxik ma'lumotlar tuzilmasiga ega bo'lamiz. Informatsiya tarmoqlarda joylashgan bo'lib, qolgan tugunlari tarmoqlarga yo'lni ko'rsatadi.

1-chi bosqich	Talaba = yozuv
2-chi bosqich	Raqam
2- chi bosqich	Ism = yozuv
3- chi bosqich	Familiya
3- chi bosqich	Ism
3- chi bosqich	Otasinig ismi
2- chi bosqich	Anket ma'lumotlari = yozuv
3- chi bosqich	Tug'ilgan joyi
3- chi bosqich	Tug'ilgan yili
3- chi bosqich	Ota-onasi = yozuv
4- chi bosqich	Onasi
4- chi bosqich	Otasi
42- chi bosqich	Fakul'tet
2- chi bosqich	Guruh
2- chi bosqich	Baholar =yozuv
3- chi bosqich	CHet tili
3- chi bosqich	Fizika

Ushbu tuzilma ichma-ich joylashgan yozuv deb ataladi.

Yozuv ustidagi amallar:

Yozuv maydoni ma'lumotlarni o'qish.

Yozuv maydoniga informatsiya kiritish.

Turga mos keluvchi, yozuv maydoni ustida bajarishi mumkin bo'lgan barcha amallar.

4. Jadvallar

Jadval - bu yozuvning chekli majmuasidir.

N	FISH	GUR UH	FAK ULTET	ANG L	FIZI KA
1					
2					
...					
n					

Jadval berilayotganda unda ishtirok etadigan yozuvlar soni ko'rsatib o'tiladi.

Masalan:

Type ST = Record

Num: Integer;

Name: String[15];

Fak: String[5];

Group: String[10];

Angl: Integer;

Physic: Integer;

var

Table: Array [1..19] of St;

Jadval ma'lumotlari elementi yozuv hisoblanadi. SHuning uchun jadval ustida bajariladigan amallar bu yozuv ustida bajariladigan amallardir.

Jadval ustida bajariladigan amallar:

1. Berilgan kalit bo'yicha yozuvni qidirish.
2. Jadvalga yangi yozuvni kiritish.

Kalit – bu yozuv identifikatori. Ushbu identifikatorni saqlash uchun maxsus maydon ajratiladi.

Qo'shma kalit – bu shunday kalitki, u ikkidan ortiq maydonni o'z ichiga oladi.

Nazorat savollari:

1. Qaysi statik tuzilma eng oddiy xisoblanadi?
2. Vektor deb nimaga aytiladi?
3. YOzuv deganda nimani tushunasiz?
4. YOzuvni e'lon qilish qanday amalga oshiriladi
5. Jadvalni asosiy elementlarini sanab bering.
6. Ularning asosiy xususiyatlarini aytib bering.
7. Statik turdagi ma'lumotlar tuzilmasi ustida bajarilishi mumkin bo'lgan amallar.

3-Ma'ruza: Yarim statik ma'lumotlar tuzilmalari.

Reja:

1. Ro'yxat
2. Stek
3. Navbat
4. Dek.

O'quv maqsadi:

Ta'limiy: Talabalarga ro'yxat, stek, navbat va dek ma'lumotlar tuzilmasi haqida ma'lumot berish.

Tarbiyaviy: Talabalarni vatanparvarlik ruhida tarbiyalash.

Rivojlantiruvchi: Talabalarning aqliy faoliyatini rivojlantirish.

1. Ro'yxat

Yarimstatik ma'lumotlar tuzilmasini o'rganishdan oldin quyidagi tushunchalar bilan tanishib chiqamiz. Yarimstatik ma'lumotlar tuzilmasiga stek, dek va navbatlar kiradi. Ro'yxat bu shunday ma'lumotlar majmuasiki, uning elementlari bog'langan bo'lib, ular turli turlarga tegishli bo'lishi mumkin.

Ro'yxatga misol:

$E_1, E_2, \dots, E_n, \dots$ $n > 1$ bo'lib n fiksirlanmagan.

Ro'yxat elementlari soni dastur bajarilishi davomida o'zgarib turishi mumkin.

Ro'yxatning 2 turi mavjud:

Bog'lanmagan

Bog'langan

Ro'yxatning bog'lanmagan turida uning elementlari orasidagi bog'liqlik oshkormas (noaniq) ko'rinishda bo'ladi. Bog'langan turida esa ma'lumot elementlariga ro'yxatda o'zidan oldingi yoki keyingi keluvchi element bilan aloqasini bildiruvchi ko'rsatich kiritiladi.

Stek, dek va navbatlar bular bog'lanmagan ro'yxatlarga misol bo'ladi. Bundan tashqari ular ketma-ket ro'yxatga misol bo'lib, oshkormas bog'liqlik ularning ketma-ketligi orqali aks etadi.

Kundalik xayotda deyarli har kuni har bir inson navbat tushunchasi bilan duch keladi. Umuman olganda navbat elementi qandaydir xizmat ko'rsatishga buyurtma bo'lib xisoblanadi: masalan, ma'lumotlar byurosidan kerakli ma'lumotni olish, kinoteatrlarda chipta olish, do'konda xarid qilib olingan mahsulotlarga kassada pul to'lash va boshq.

Dasturlashda shunday ma'lumotlar tuzilmasi mavjudki, u navbat deb ataladi. Bu turdagi ma'lumotlar tuzilmasida kelib tushgan buyurtmalarga xizmat ko'rsatish tartibi aniqlanadi.

Navbatlar yarimstatik tuzilma xisoblanib, vaqt o'tishi va navbat uzunligiga qarab, uni tashkil etuvchi elementlar o'zgarib turishi mumkin.

Navbatni tashkil qiluvchi elementlarga xizmat ko'rsatilishiga qarab, navbatning asosiy ikkita ko'rinishi mavjud:

Navbatning birinchi ko'rinishida, navbatga kelib tushgan birinchi elementga birinchi bo'lib xizmat ko'rsatiladi va navbatdan chiqariladi. Mazkur ko'rinishdagi xizmat ko'rsatishni FIFO (First input-First output, ya'ni birinchi kelgan – birinchi ketadi) nomlash qabul qilingan. Navbat har ikkala tomondan ochiq bo'ladi.

2. Ikkinchi ko'rinishni LIFO (Last input - First output, ya'ni oxirgi kelgan – birinchi ketadi) deyilib, navbatga kelib tushgan oxirgi buyurtma (element)ga birinchi bo'lib xizmat ko'rsatiladi. Mazkur ko'rinishdagi navbatni dasturlashda STEK deb nomlash qabul qilingan.

2. Stek

LIFO, ya'ni navbatning oxirgi bo'lib kirgan elementiga birinchi bo'lib xizmat ko'rsatiladi. Bu eng ko'p ishlatiladigan ma'lumotlar tuzilmalaridan biri bo'lib, turli xil masalalarni hal qilishda ancha qulay va samarali xisoblanadi.

```
* // Stack abstract class  
template <typename E> class Stack {  
private:  
void operator =(const Stack&) {} // Protect assignment  
Stack(const Stack&) {} // Protect copy constructor  
public:  
Stack() {} // Default constructor  
virtual ~Stack() {} // Base destructor  
// Reinitialize the stack. The user is responsible for  
// reclaiming the storage used by the stack elements.  
virtual void clear() = 0;  
// Push an element onto the top of the stack.  
// it: The element being pushed onto the stack.  
virtual void push(const E& it) = 0;  
// Remove the element at the top of the stack.  
// Return: The element at the top of the stack.  
virtual E pop() = 0;  
// Return: A copy of the top element.  
virtual const E& topValue() const = 0;  
// Return: The number of elements in the stack.  
virtual int length() const = 0;
```

```
};
```

Figure 4.17 The stack ADT.

Xizmat ko'rsatishni keltirilgan tartibiga ko'ra, stekda faqatgina bitta pozitsiyaga murojaat s

3. Navbat

Dasturlashda shunday ma'lumotlar tuzilmasi mavjudki, u navbat deyiladi. Bunday ma'lumotlar tuzilmasi real navbatni modellashtirishda katta ahamiyatga ega. Bunda xizmat ko'rsatishga kelib tushgan talab, uning ijrosi, ya'ni xizmat ko'rsatish tartibini aniqlashda zarur bo'ladi. Kundalik hayotimizdan barchamizga ma'lum bo'lgan navbat turi, dasturlashda FIFO (First input-First output, ya'ni birinchi kelgan – birinchi ketadi) deb nomlanadi.

```
// Operation choices: DOMOVE will move a disk  
// DOTOH corresponds to a recursive call  
enum TOHop { DOMOVE, DOTOH };  
class TOHobj { // An operation object  
public:  
TOHop op; // This operation type  
int num; // How many disks  
Pole start, goal, tmp; // Define pole order  
// DOTOH operation constructor  
TOHobj(int n, Pole s, Pole g, Pole t) {  
op = DOTOH; num = n;  
start = s; goal = g; tmp = t;  
}  
// DOMOVE operation constructor  
TOHobj(Pole s, Pole g)  
{ op = DOMOVE; start = s; goal = g; }
```

```

};
void TOH(int n, Pole start, Pole goal, Pole tmp,
Stack<TOHObj*> & S) {
S.push(new TOHObj(n, start, goal, tmp)); // Initial
TOHObj* t;
while (S.length() > 0) { // Grab next task
t = S.pop();
if (t->op == DOMOVE) // Do a move
move(t->start, t->goal);
else if (t->num > 0) {
// Store (in reverse) 3 recursive statements
int num = t->num;
Pole tmp = t->tmp; Pole goal = t->goal;
Pole start = t->start;
S.push(new TOHObj(num-1, tmp, goal, start));
S.push(new TOHObj(start, goal));
S.push(new TOHObj(num-1, start, tmp, goal));
}
delete t; // Must delete the TOHObj we made
}
}
}

```

Figure 4.22 Stack-based implementation for Towers of Hanoi

Bu erdan ko'rinib turibdiki, stekdan farqli ravishda xizmat ko'rsatilish birinchi kelgan elementga birinchi bo'lib xizmat ko'rsatiladi. Stekdan yana bir farqi, bunda navbatning har ikkala tomoni ochiq bo'ladi, ya'ni bir tomondan kelib ikkinchi tomondan chiqib ketadi.

Demak, navbatda elementni olish ro'yxat boshidan, yozish esa oxiridan amalga oshiriladi.

EHM xotirasida real navbat elementlari soni chekli bo'lgan bir o'lchamli massiv ko'rinishida yaratiladi. Albatta, bunda navbat elementi turini ko'rsatish va

navbat bilan ishlashni ko'rsatuvchi o'zgaruvchi zarur bo'ladi.

Navbat fizik bosqichda xotira sohasini ro'yxat ketma-ketligi bo'yicha to'laligicha egallaydi.

Navbat ustida amalga oshiriladigan amallar:

Navbat uchun 3 ta oddiy amal aniqlangan.

Navbatga yangi element joylashtirish: insert (y,x), bu erda q-navbat, x – element.

Navbat boshidan elementni o'chirish: remove(y)

Navbatni bo'sh yoki bo'sh emasligini aniqlash: empty (y)

Bundan tashqari, navbat bir o'lchamli massiv ko'rinishida ifodalanganligi uchun massivni to'la yoki to'la emasligini kuzatib turish lozim bo'ladi. SHu maqsadda, full(q) amali kiritiladi.

Umuman olganda, insert amalini har doim bajarish mumkin. Sababi, navbatni tashkil qiluvchi elementlar soniga cheklanishlar qo'yilmagan. Remove amali esa faqatgina navbat bo'sh bo'lmagandagina ishlaydi. Empty amali esa har doim o'rinli.

C++ tilida navbatni bir o'lchamli massiv ko'rinishda amalga oshirishga misol:

```
// Abstract queue class  
template <typename E> class Queue {  
private:  
void operator =(const Queue&) {} // Protect assignment  
Queue(const Queue&) {} // Protect copy constructor  
public:  
Queue() {} // Default  
virtual ~Queue() {} // Base destructor  
// Reinitialize the queue. The user is responsible for  
// reclaiming the storage used by the queue elements.  
virtual void clear() = 0;  
// Place an element at the rear of the queue.  
// it: The element being enqueued.
```

```

virtual void enqueue(const E&) = 0;
// Remove and return element at the front of the queue.
// Return: The element at the front of the queue.
virtual E dequeue() = 0;
// Return: A copy of the front element.
virtual const E& frontValue() const = 0;
// Return: The number of elements in the queue.
virtual int length() const = 0;
};

```

Nazorat savollari:

1. Navbat bilan stek qanday ma'lumotlar tuzilmasiga kiradi?
2. Stekdan elementni tanlash qanday amalga oshiriladi?
3. Stekni yuqori elementini o'chirmasdan o'qish qay yo'sinda amalga oshiriladi?
4. Qanday xizmat ko'rsatish turiga FIFO, qaysi biriga LIFO deb ataladi?
5. Halqasimon navbatining to'laganligi belgisi nimadn iborat?
6. Navbatning bo'shligi belgisi?
7. Ro'yxat deb nimaga aytiladi?
8. Ro'yxat turlarini aytib o'ting.
9. Navbat elementlarini sanab o'ting.
10. Halqasimon navbat qanday tashkil qilinadi?
11. Dekning o'ziga xosligi nimadan iborat?

4-Ma'ruza. Graflar va daraxtlar

Reja:

1. Rekursiya xaqida tushuncha.
2. Daraxtlar. Ularni tasvirlash.
3. Binar daraxtlar.
4. Ko'p o'lchamli daraxtni binar ko'rinishga keltirish.

O'quv maqsadi:

Ta'limiy: Talabalarda “Axborot xavfsizligini ta'minlashning usul va vositalari” mavzusining mohiyati, maqsadi va vazifalari, hamda axborot xavfsizligining hozirgi kundagi o'rnini va rivojlanish istiqbollari, ularning fan-texnika taraqqiyotida, jamiyat rivojida tutgan o'rnini to'g'risida bilim va ko'nikmalarni shakllantirish.

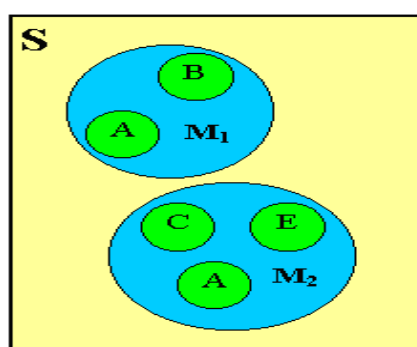
Tarbiyaviy: Axborot xavfsizligi, ularning ahamiyatini tushuntirish orqali talabalarning qiziqishini oshirish, axborotdan foydalanishda xavfsizlik ko'nikmalarini shakllantirish, ilmga intiluvchanlik ruhida tarbiyalash.

Rivojlantiruvchi: Talabalarning “Axborot xavfsizligini ta'minlashning usul va vositalari” mavzusidan olgan bilimlarini mutaxassislik sohasidagi ishlarida qo'llash orqali rivojlantirish.

Rekursiv algoritmlar va rekursiv ma'lumotlar tuzilmalarini ko'rib chiqaylik.

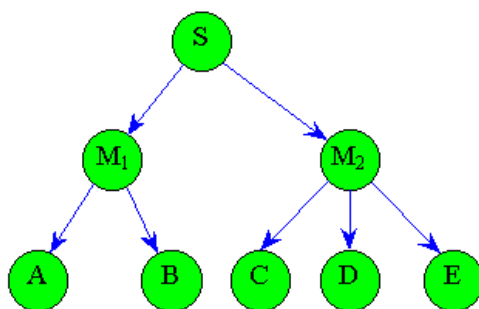
Rekursiya – shunday jarayonki, bunda jarayonni borishi o'ziga murojaat qilish bilan bog'liq bo'ladi.

Rekursiv ma'lumotlar tuzilmasiga misol qilib shunday tuzilmalarni olish mumkinki, ushbu tuzilmalarning elementlarining o'zi ham xuddi shunday tuzilma bo'ladi (qarang, chizma).



1. Daraxtlar

Daraxt – bu chiziqsiz bog'langan ma'lumotlar tuzilmasidir (qarang, chizma).



Daraxt o'zining quyidagi belgilari bilan tasniflanadi:

- daraxtda shunday bitta element borki, unga boshqa elementlardan murojaat yo'q. Mazkur elementga daraxt ildizi deyiladi;

- daraxtda ixtiyoriy elementga chekli sondagi ko'rsatkichlar yordamida murojaat qilish mumkin;

- daraxtning har bir elementi faqatgina o'zidan oldingi kelgan bitta element bilan bog'langan. Daraxtning har bir tuguni oraliq yoki terminal (barg) bo'lishi mumkin. YUqoridagi chizmada M1, M2 - oraliq, A, B, C, D, E - barglardir. Terminal tugunning o'ziga xos tasnifi uning shoxlari yo'qligidir.

Balandlik – bu daraxt bosqichi soni. YUqoridagi chizmadagi daraxt balandligi ikkiga teng.

Daraxt tugunlaridan chiqayotgan shoxlar soni tugundan chiqish darajasi deyiladi (Keltirilgan chizmada M1 uchun chiqish darajasi 2, M2 uchun esa 3 ga teng). Daraxtlar chiqish darajasi bo'yicha sinflarga ajratiladi:

- 1) agar maksimal chiqish darajasi m bo'lsa, u holda bunday daraxt m -chi tartibli daraxt deyiladi;

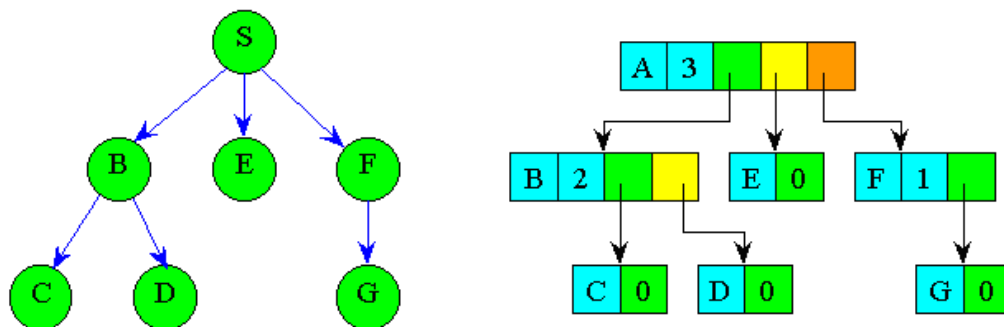
- 2) agar chiqish darajasi 0 yoki m bo'lsa, u holda to'liq m -chi tartibli daraxt bo'ladi;

- 3) agar maksimal chiqish darajasi 2 bo'lsa, u holda bunday daraxt binar daraxt deyiladi;

- 4) agar chiqish darajasi 0 yoki 2 bo'lsa, u holda to'liq binar daraxt deyiladi.

Tugunlar orasidagi bog'liqlikni tavsiflash uchun yana quyidagicha termindan foydalaniladi: M1 – A va V elementlar uchun “ota” . A va V – esa M1 tugun “o'g'illari”.

Daraxtlarni tasvirlash



Daraxtni grafik shakldagi va uning chiziqsiz ro'yxat shaklidagi ifodalanishi

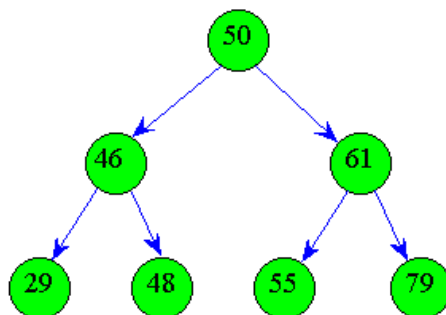
EXM xotirasida daraxtni ifodalashaning eng qulay usuli bu uni bog'langan ro'yxatlar ko'rinishida ifodalashdir. Ro'yxat elementi tugun qiymati va chiqish darajasini o'z ichiga oluvchi informatsion maydonga xamda chiqish darajasiga teng bo'lgan ko'rsatkichlar maydoniga ega bo'lishi lozim (yuqoridai chizma), ya'ni elementning har bir ko'rsatkichi ushbu elementni tugun o'g'illari bo'lgan tugunlarga yo'nalishini aniqlaydi.

2. Binar daraxtlar

Binar daraxtlar eng ko'p foydalaniladigan daraxtlar turi xisoblanadi.

Daraxtlarni EXM xotirasida tasvirlanishiga ko'ra xar bir element to'rtta maydonga ega yozuv xisoblanadi. Mazkur maydonlar qiymati mos ravishda yozuv kaliti bo'lib, boshqa elementlarga murojaatni ifodalaydi, ya'ni chapga-pastga, o'nga-pastga va yozuv matniga.

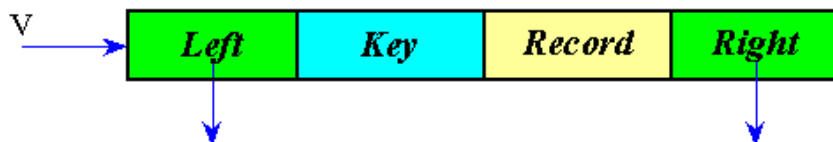
Shuni esda tutish lozimki, daraxt xosil qilinayotganda, otaga nisbatan chap tomondagi o'g'il qiymati kichik kalitga, o'ng tomondagi o'g'il esa katta qiymatli kalitga ega bo'ladi. Masalan, quyidagi elementlardan binar daraxt quramiz: 50, 46, 61, 48, 29, 55, 79. U quyidagi ko'rinishga ega bo'ladi:



Natijada, o'ng va chap qism daraxtlari bir xil bosqichli tartiblangan binar daraxt

xosil qildik. Agar daraxtning o'ng va chap qism daraxtlari bosqichlari farqi birdan kichik bo'lsa, bunday daraxt ideal muvozanatlangan daraxt deyiladi. YUqorida xosil qilgan binar daraxtimiz ideal muvozanatlangan daraxtga misol bo'ladi.

Binar daraxtni xosil qilish uchun EXM xotirasida elementlar quyidagi turda bo'lishi lozim:



$V = \text{MakeTree}(\text{Key}, \text{Rec})$ amali ikkita ko'rsatkichli (kalit) va ikkita maydonli (informatsion) element yaratadi (daraxt tuguni)

MakeTree protsedursi ko'rinishi:

Paskal'

New(y);

$p^{\wedge}.r := \text{rec};$

$p^{\wedge}.k := \text{key};$

$v := p;$

$p^{\wedge}.\text{left} := \text{nil};$

$p^{\wedge}.\text{right} := \text{nil};$

Boshida kalit birinchi qiymati kiritiladi. Undan so'ng elementni o'zini maketree protsedurasi orqali hosil qilamiz. Keyin esa ko'rsatkich bo'sh qiymatni ko'rsatguncha tsiklni davom ettiramiz.

READ(key,rec)

tree=maketree(key,rec)

WHILE not eof DO

READ(key,rec)

V=maketree(key,rec)

WHILE P<>nil DO

Q=P

```

IF key=k(P)
THEN P=left(P)
ELSE P=right(P)
END IF
END WHILE
IF P=nil
THEN WRITELN(' Bu ildiz');
tree=V
ELSE IF key<k(q)
    THEN left(P)=V
    ELSE right(P)=V
END IF
END IF
END WHILE

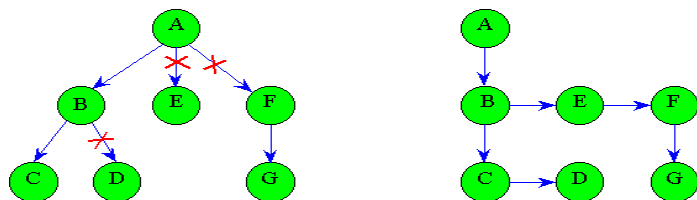
```

4. m-o'lchamli daraxtni binar ko'rinishga keltirish

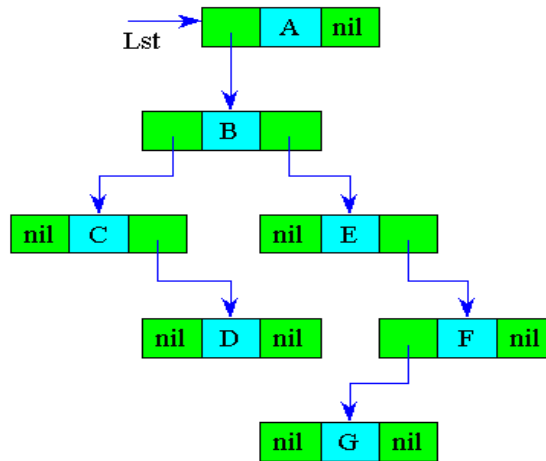
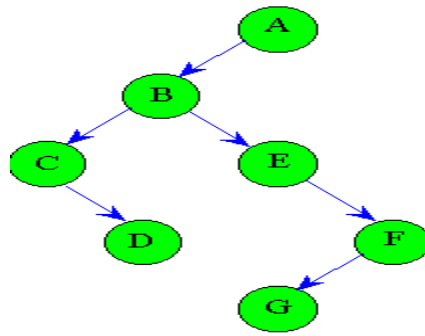
Noformal algoritim:

1. Daraxtning xar bir tugunida katta o'g'ilga mos chetki chap shoxidan tashqari barcha shoxlari kesib tashlanadi.
2. Bitta ota barcha o'g'illari gorizontaal chiziq bilan ulanadi.
3. Xosil qilingan tuzilmaning xar bir tugunida katta o'g'il mazkur tugun pastida turgan tugun xisoblanadi (agar u mavjud bo'lsa).

Algoritim amallar ketma-ketligi quyida keltirilgan.



yoki



m-o'lchovli daraxtni binar ko'rinishga keltirish.

5. Daraxtlar ustida bajariladigan amallar

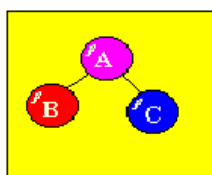
1. Daraxt ko'ruvi (Obxod dereva).
2. Qism daraxtni o'chirish.
3. Qism daraxt qo'yish.

Daraxt ko'ruvini amalga oshirish uchun quyidagi uchta protsedurani bajarish

lozim:

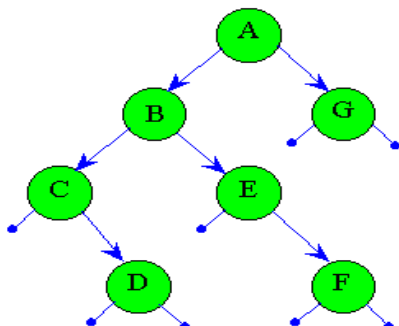
1. Ildizni qayta ishlash.
2. CHap tarmoq(shox)ni qayta ishlash.
3. O'ng tarmoq(shox)ni qayta ishlash.

YUqoridagi protsedura qanday ketma-ketlikda amalga oshirilishiga qarab ko'ruvni uchta ko'rinishga ajratiladi.



1. YUqoridan quyiga. Protsedur quyidagi ketma-ketlikda bajariladi A-B-C.
2. CHapdan o'ng. Protsedur quyidagi ketma-ketlikda bajariladi B-A-C.
3. Quyidan yuqoriga. Protsedur quyidagi ketma-ketlikda bajariladi B-C-A.

Masalan quyidagi daraxtda ko'ruv o'tkazaylik.



Daraxat ko'ruvi tartibi:

YUqoridan pastga:

A,B,C,D,E,F,G.

CHapdan o'nga:

C,D,B,E,F,A,G.

Pastdan yuqoriga:

D,C,F,E,B,G,A.

Daraxt ko'rigini rekursiv protsedurlari:

1) PROCEDURE pretrave (tree) {yuqoridan pastga ko'rik}

```

IF tree<>nil
  THEN PRINT info (tree)
        pretrave (left (tree))
        pretrave (right (tree))
END IF
RETURN
  
```

2) PROCEDURE intrave (tree)

```

IF tree<>nil
  THEN intrave (left (tree))
        PRINT info (tree)
        intrave (right (tree))
END IF
RETURN
  
```

3) PROCEDURE postrave (tree)

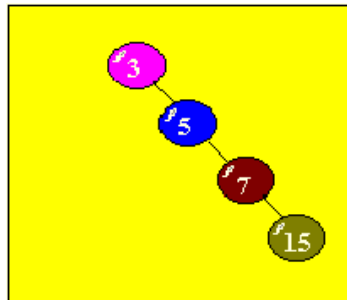
```

IF tree<>nil
  THEN postrave (left (tree))
    postrave (right (tree))
  PRINT info (tree)
END IF
RETURN

```

Binar daraxt bo'yicha qidiruv protsedurasi

Mazkur protseduraning vazifasi shundan iboratki, u berilgan kalit bo'yicha daraxt tuguni qidiruvini amalga oshiradi. Qidiruv operatsiyasining davomiyligi daraxt tuzilishiga bog'liq bo'ladi. Haqiqatdan, agar elementlar daraxtga kalit qiymatlari o'sish (kamayish) tartibida kelib tushgan bo'lsa, u holda daraxt bir tomonga yo'nalgan ro'yxat hosil qiladi (chiqish darajasi bir bo'ladi, ya'ni yagona shohga ega), masalan:



Bu holda daraxtda qidiruv vaqti, bir tomonlama yo'naltirilgan ro'yxatdagi kabi bo'lib, o'rtacha qarab chiqishlar soni $N/2$ bo'ladi.

Agar daraxt muvozanatlangan bo'lsa, u holda qidiruv eng samarali natija beradi. Bu holda qidiruv $\log_2 N$ dan ko'p bo'lmagan elementlarni ko'rib chiqadi.

Qidiruv protsedurasini ko'rib chiqamiz. search o'zgaruvchisiga topilgan bo'g'in ko'rsatkichi o'zlashtiriladi:

```

p=tree
WHILE p<>nil DO
  IF key=k (y)
    THEN search=y
  RETURN

```

```

END IF
IF key<>k (y)
    THEN p=left (y)
    ELSE p=right (y)
END IF
END WHILE
search=nil
RETURN

```

Daraxtga yangi element qo'shish protsedurasi

Daraxtga biror bir elementni qo'shishdan oldin daraxtda berilgan kalit bo'yicha qidiruvni amalga oshirish lozim bo'ladi. Agar berilgan kalitga teng kalit mavjud bo'lsa, u holda dastur o'z ishini yakunlaydi, aks holda daraxtga element qo'shish amalga oshiriladi.

Daraxtga yangi yozuvni kiritish uchun, avvalo daraxtni shunday tugunini topish lozimki, natijada mazkur tugunga yangi element qo'shish mumkin bo'lsin. Kerakli tugunni qidirish algoritmi ham xuddi berilgan kalit bo'yicha tugunni topish algoritmi kabi bo'ladi. Biroq berilgan kalit bo'yicha qidiruv protsedurasidan to'g'ridan-to'g'ri (bevosita) foydalanib bo'lmaydi, sababi, qidiruv protsedurasida, qaysi tugunda murojaat NIL (search = nil) bo'lgani fiksirlanmaydi.

Qidiruv protsedurasini shunday modifikatsiya qilamizki, qo'shimcha samara sifatida yangi protseduramiz berilgan kalit turgan tugunni fiksirlasin (qidiruv muvofaqiyatli bo'lsa), yoki shunday tugunni, ushbu tugunni qayta ishlagandan keyin qidiruv yakunlansin (qidiruv muvofaqiyatli bo'lsa).

Daraxtda qo'shilayotgan element kalitiga teng kalitli element yo'q bo'lgan holda elementni qo'shish protsedurasini keltirib o'tamiz.

```

q=nil
p=tree
WHILE p<>nil DO
    q=y
    IF key=k (y)

```

```

    THEN search=p
    RETURN
END IF
IF key<k (y)
    THEN p=left (y)
    ELSE p=right (y)
END IF
END WHILE

```

{Berilgan kalitga teng tugun topilmadi, element qo'shish talab qilinadi. Ota bo'lishi mumkin tugunga q ko'rsatkich beriladi.}

```
V=maketree (key, rec)
```

{Qo'yilayotgan V element chap yoki o'ng o'g'il bo'lishini aniqlash lozim.}

```

IF key<k (y)
    THEN left (y)=V
    ELSE right (y)=V
END IF

```

```
search=V
```

```
RETURN
```

Binar daraxtdan elementni o'chirish protsedurasi

Tugunni o'chirib tashlash natijasida daraxtning tartiblanganligi buzilmasligi lozim. Tugun daraxtdan o'chirilayotganda 3 xil variant bo'lishi mumkin:

1) Topilgan tugun terminal (barg). Bu holatda tugun shunchaki o'chirib tashlanadi.

2) Topilgan tugun faqatgina bitta o'g'ilga ega. U holda o'g'il ota o'rniga joylashtiriladi.

3) O'chirilayotgan tugun ikkita o'g'ilga ega. Bunday holatda shunday qism daraxtlar zvenosini topish lozimki, uni o'chirilayotgan tugun o'rniga qo'yish mumkin bo'lsin. Bunday zveno har doim mavjud bo'ladi:

- bu yoki chap qism daraxtning eng o'ng tomondagi elementi (ushbu zvenoga

erishish uchun keyingi uchiga chap shoh orqali o'tib, navbatdagi uchlariga esa, murojaat NIL bo'lmaguncha, faqatgina o'ng shohlari orqali o'tish zarur).

- yoki o'ng qism daraxtning eng chap elementi (ushbu zvenoga erishish uchun keyingi uchiga o'ng shoh orqali o'tib, navbatdagi uchlariga esa, murojaat NIL bo'lmaguncha, faqatgina chap shohlari orqali o'tish zarur).

O'chirilyotgan element chap qism daraxtining eng o'ngidagi element o'chirilyotgan element uchun "Predshestvennik" bo'ladi (12 uchun – 11 bo'ladi). Merosxo'r esa o'ng qism daraxtning eng chapidagi tuguni (12 uchun - 13).

Merosxo'rni topish algoritmini ishlab chiqaylik (5.9 chizma).

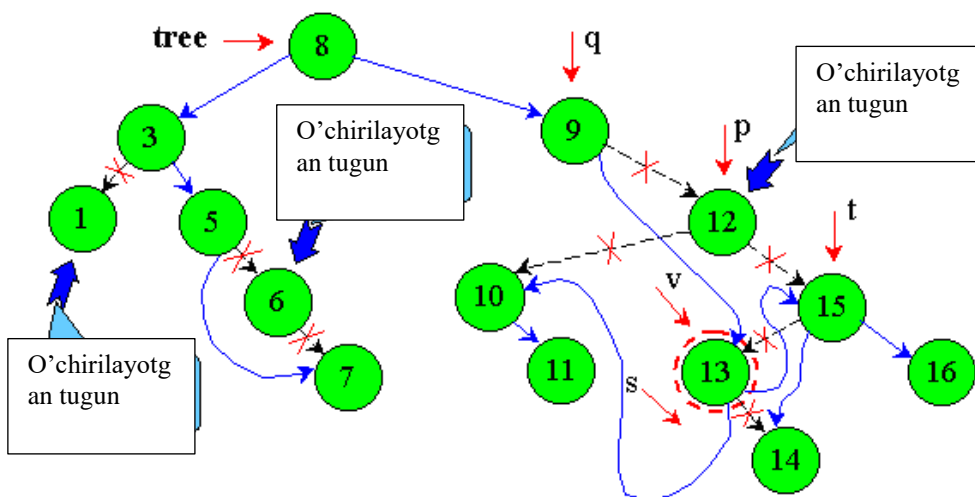
p – ishchi ko'rsatkich;

q - r dan bir qadam orqadagi ko'rsatkich;

v – o'chirilyotgan tugun merosxo'rini ko'rsatadi;

t - v bir qadam orqada yuradi;

s - v dan bir qadam oldinda yuradi (chap o'g'ilni yoki bo'sh joyni ko'rsatib boradi).



Yuqoridagi daraxt bo'yicha qaraydigan bo'lsak, oxir oqibatda, v ko'rsatkich 13 tugunni, s esa bo'sh joyni ko'rsatishi lozim.

1) Elementni qidirish protsedurasi orqali o'chirilyotgan elementni topamiz. r ko'rsatkich o'chirilyotgan elementni ko'rsatadi.

2) O'chiriladigan elementni o'rniga qo'yiluvchi tugunga v ko'rsatkich qo'yamiz.


```

IF left (y)=nil
  THEN v=right (y)
  ELSE IF right (y)=nil
    THEN v=left (y)
    ELSE t=y
      v=right (y)
      s=left (y)
WHILE s<>nil
  t=v
  v=s
  s=left (v)
END WHILE
IF t<>p
  THEN WRITE (v element p ning o'g'li emas)
    left (t)=right (v)
    right (v)=right (y)
  END IF
left (v)=left (y)
IF q=nil
  THEN WRITE (v ildiz)
    tree=v
  RETURN
  END IF
IF p=left (y)
  THEN left (y)=v
  ELSE right (y)=v
  END IF
END IF
END IF
END IF
FREENODE (y) (ushbu protsedura bo'sh tugun hosil qiladi, ya'ni o'chirilgan

```

element joylashgan xotira yacheykasini tozalaydi.)

RETURN

Nazorat savollari:

- 1.Rekursiya nima?
- 2.Daraxt nima? Uning o'ziga xos xususiyatlarini aytib bering.
- 3.To'liq daraxt deganda nimani tushunasiz?
- 4.Daraxt ko'ruvi nimadan iborat?
- 5.Har qanday daraxtni binar ko'rinishga keltirish mumkinmi?
- 6.Daraxt tuguni qanday hosil qilinadi?
- 7.Daraxtda qanday amallarni bajarish mumkin?

5-Ma'ruza:Ko'p bog'liq ro'yxat bilan aks ettiriladigan ma'lumotlar

Tuzilmalari.

Reja:

1. Bog'langan ro'yxatlar
2. Bir bog'lamli ro'yxatlar
3. Bir bog'lamli halqasimon ro'yxatlar

O'quv maqsadi:

Ta'limiy: Talabalarda bog'langan ro'yxatlar, bir bog'lamli ro'yxatlar va bir bog'lamli halqasimon ro'yxatlar to'g'risida bilim va ko'nikmalarni shakllantirish.

Tarbiyaviy: Chiziqsiz dinamik ma'lumotlar tuzilmasi orqali bajariladigan ishlar, ularning ahamiyatini tushuntirish orqali talabalarning qiziqishini oshirish, ilmga intiluvchanlik ruhida tarbiyalash;

Rivojlantiruvchi: Talabalarning "Chiziqsiz dinamik ma'lumotlar tuzilmasi" mavzusidan olgan bilimlarini mutaxassislik sohasidagi ishlarida qo'llash orqali rivojlantirish.

1. Bog'langan ro'yxatlar

Dastur bajarilayotganda vujudga keladigan yoki o'lchamlari dastur bajarilishi

mobaynida aniqlanadigan ob'ektlar dinamik ob'ektlar deyiladi.

```
// Singly linked list node
template <typename E> class Link {
public:
    E element; // Value for this node
    Link * next; // Pointer to next node in list
// Constructors
    Link(const E& elemval, Link * nextval =NULL)
    { element = elemval; next = nextval; }
    Link(Link * nextval =NULL) { next = nextval; }
};
```

Figure 4.4 A simple singly linked list node implementation.

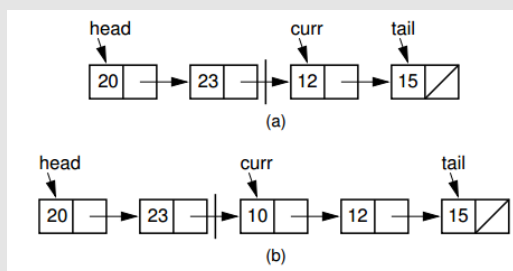


Figure 4.5 Illustration of a faulty linked-list implementation where *curr* points directly to the current node. (a) Linked list prior to inserting element with value 10. (b) Desired effect of inserting element with value 10.*

Dinamik ma'lumotlar tuzilmasi 2 ta xususiyatga ega:

1. Tuzilmada elementlar soni oldindan aniqlanmagan.
2. Dinamik tuzilma elementlari qat'iy chiziqli tartiblanmagan.

Ular xotirada tartibsiz joylashgan bo'lishi mumkin. Dinamik tuzilma elementlarini o'zaro bog'lash uchun tuzilma elementlari tarkibiga informatsion maydondan tashqari ko'rsatkichlar maydoni xam kiradi (qarang, chizma) (tuzilmani boshqa elementlari bilan bog'liqlik).

P1 va P2 – o'zaro bog'langan elementlarni adreslarini o'z ichiga oluvchi ko'rsatkichlardir. Ko'rsatkichlar slot raqamini o'z ichiga oladi.

Bog'langan ro'yxatlar eng ko'p tarqalgan dinamik tuzilmalardan xisoblanadi. Ma'lumotlarni mantiqiy tasvirlash nuqtai nazaridan ro'yxatlar ikkitaga ajratiladi: chiziqli va chiziqsiz.

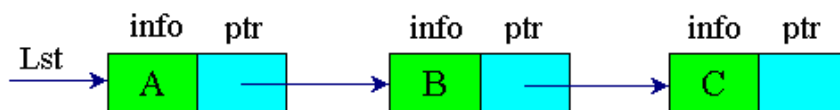
Chiziqli ro'yxatlarda elementlar orasidagi bog'liqlik qat'iy tartiblangan bo'ladi, ya'ni element ko'rsatkichi o'zidan navbatdagi element adresini o'z ichiga oladi yoki aksincha.

Chiziqli ro'yxatlarga bir va ikki bog'lamli ro'yxatlar kiradi. Chiziqsiz ro'yxatlarga esa ko'p bog'lamli ro'yxatlar kiradi.

Umuman olganda ro'yxat elementi bir yoki bir necha ko'rsatkichlar yozuvi maydonini namoyish qiladi.

2. Bir bog'lamli ro'yxatlar

Bir bog'lamli ro'yxat elementi ikkita maydonga ega (chizma): informatsion maydon (INFO) va ko'rsatkich maydoni (PTR).

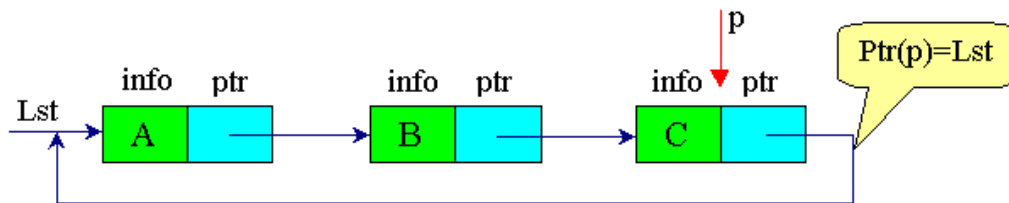


Bir bog'lamli ro'yxat ko'rsatkichining o'ziga xosligi shundan iboratki, bunda faqatgina o'zidan keyin keluvchi ro'yxat elementi adresini ko'rsatadi. Ro'yxat eng so'ngi elementining ko'rsatkich maydoni bo'sh bo'ladi (NIL). LST – ro'yxat boshi ko'rsatkichi. Umuman olganda ro'yxat bo'sh ham bo'lishi mumkin, bu holda LST NIL bilan ustma-ust tushadi, ya'ni teng bo'ladi.

Ro'yxat elementiga murojat faqatgina ro'yxat boshidan amalga oshiriladi, ya'ni bu ro'yxatda teskari aloqa yo'q.

3. Halqasimon bir bog'lamli ro'yxat

Halqasimon bir bog'lamli ro'yxat oddiy bir bog'lamli ro'yxatda eng so'ngi element ko'rsatkichiga ro'yxat boshi elementi ko'rsatkichi qiymatini o'zlashtirish orqali hosil qilinadi (chizma).



Bir bog'lamli ro'yxatlar ustida bajariladigan oddiy amallar

Bir bog'lamli ro'yxat boshiga element qo'yish.

Faraz qilaylik bir bog'lamli ro'yxat boshiga informatsion maydoni D bo'lgan element qo'yish talab qilingan bo'lsin. Buning uchun bo'sh element kiritish lozim ($P = \text{GetNode}$). Ushbu elementning informatsion maydoniga D ($\text{INFO}(P) = D$)ning qiymati o'zlashtiriladi, ko'rsatkich qiymatiga esa ro'yxat boshini anglatuvchi ko'rsatkich qiymati o'zlashtiriladi ($\text{Ptr}(P) = \text{Lst}$), ro'yxat boshi ko'rsatkichi qiymatiga esa P ko'rsatkich qiymati o'zlashtiriladi ($\text{Lst} = P$).

Paskal tilida amalga oshirish quyidagicha:

type

PNode = TNode;

TNode = record

Info: Integer; {ro'yxat elementlari turi – ixtiyoriy bo'lishi mumkin}

Next: PNode;

end;

var

Lst: PNode; {ro'yxat boshini ko'rsatuvchi ko'rsatkich}

P: PNode;

Boshiga qo'yish

New(P); {yangi element yaratish}

$P^{\wedge}.\text{Info} := D$;

$P^{\wedge}.\text{Next} := \text{Lst}$; {P ro'yxat boshni ko'rsatadi, lekin Lst P ni ko'rsatmaydi – yangi boshlanish}

$\text{Lst} := P$; {Lst ro'yxatning yangi boshini ko'rsatadi }

Bir bog'lamli ro'yxatning boshidan elementni o'chirib tashlash.

Faraz qilaylik, ro'yxatning birinchi elementini o'chirib, lekin ushbu element

to'g'risidagi ma'lumotni Info maydonida saqlab qolish talab qilinsin. Buning uchun o'chirilayotgan elementni ko'rsatuvchi P ko'rsatkich kiritib olamiz ($P = Lst$). X o'zgaruvchiga o'chirilayotgan element informatsion maydonining qiymatini beramiz ($X = Info(P)$). Ro'yxat boshini ko'rsatuvchi ko'rsatkichga navbatdagi element ko'rsatkichi o'zlashtiriladi ($Lst = Ptr(P)$) xamda elementni o'chiramiz ($FreeNode(P)$).

Paskal tilida amalga oshirish quyidagicha:

Elementni ro'yxatdan o'chirish

$P := Lst;$

$X := P.^{Info};$

$Lst := P.^{Next};$

$Dispose(P);$ {Element dinamik xotiradan o'chiriladi}

6-ma'ruza: Ma'lumotlarni tartibga solish tushunchasi. Ma'lumotlar tuzilmasini saralash usullari.

Reja:

1. Saralash tushunchasi, uning turlari.
2. To'g'ridan to'g'ri qo'shish orqali saralash.
3. To'g'ridan to'g'ri tanlash orqali saralash.

O'quv maqsadi:

Ta'limiy: Talabalarga saralash va uning ko'rinishlari haqida tushuncha va malaka hosil qilish.

Tarbiyaviy: Saralash orqali bajariladigan ishlar, ularning ahamiyatini tushuntirish orqali talabalarni bu fanga qiziqish ruhida tarbiyalash;

Rivojlantiruvchi: Talabalarning "Ma'lumotlarni saralash usullari" mavzusidan olgan bilimlarini mutaxassislik sohasidagi ishlarida qo'llash orqali rivojlantirish.

Saralash – bu berilgan to'plam elementlarini biror bir tartibda joylashtirish jarayonidir. Saralashni maqsadi tartiblangan to'plamda kerakli elementni topishni osonlashtirishdan iborat. Saralash dasturlarni translyatsiya qilinayotganda,

ma'lumotlar majmuasini tashqi xotirada tashkil qilinayotganda, kutubxonalar, kataloglar, ma'lumotlar bazasi yaratilayotganda tadbiiq qilinadi. Ma'lumki, saralashning turli hil algoritmlari mavjud. Sababi, bitta masalani saralash uchun juda ko'plab turli hil algoritmlardan foydalanish mumkin. Berilgan masalani hal qilishda ba'zilar mukammal bo'lishi mumkin. SHuning uchun saralash masalasida algoritmlarni qiyosiy tahlilini o'tkazish zarurati paydo bo'ladi.

Bizga X fayl berilgan bo'lsin. Fayl

$$\lambda(1), \lambda(2), \dots, \lambda(n) \quad (1)$$

yozuvlardan tashkil topgan. Har bitta $\lambda(i)$ yozuvga qandaydir xossa, boshqacha aytganda Kod $\lambda(i)$ ($i=1, n$) kalit berkitilgan deb hisoblaymiz. Odatda kalit - bu qandaydir alohida yozuv sohasi yoki yozuv sohalari kombinasiyasidir. Ushbu kalitlar to'plami elementlari kamaymaslik (o'smaslik) tartibida joylashtirilishi mumkin, deb hisoblansin.

Faylni saralash masalasining qo'yilishi: (1) yozuvlarning shunday ketma-ketlik kombinatsiyasi topilsinki, ularning kalitlari kamaymaslik tartibida joylashsin:

$$\mu(1), \mu(2), \dots, \mu(n)$$

$$K_{\mu(1)} \leq K_{\mu(2)} \leq \dots \leq K_{\mu(n)} \quad (2)$$

Ilmiy-texnik masalalarni echishda yozuv ko'pincha kalit sohasidan iborat bo'ladi. (1) fayldan (2) faylni hosil qilish uchun EHM xotirasida fayllarning fizik jihatdan o'rin almashinuvi talab etiladi. Ko'p hollarda (2) o'rin almashinuvni real holda olish talab etilmaydi. Bunda (2) ni u yoki bu usul bilan shunday tavsiflash kerakki, (1) yozuvlarga bevosita murojaat ularning kalitlari ketma-ketligi tartibida amalga oshirilsin. Buni masalan, fayldagi (2) elementlar adreslari ro'yxatini tuzish yo'li bilan amalga oshirish mumkin. (1) uchun bunday ro'yxatni tuzish adresli saralash deb ataladi.

1-Misol. Quyidagi jadvalda 7 ta yozuvdan iborat fayl keltirilgan. Ularning har biri simvulli massivning bitta elementidan iborat bo'ladi. YOzuv alohida 5 ta sohadan iborat: nomer, familiya-ism, kurs, fan, olgan bahosi. Ushbu faylni alfavit tartibida adresli kodlash quyidagi ro'yxatni tuzish bilan amalga oshiriladi:

3,5,6,7,2,4,1. (3)

	F.I.SH	Kurs	Fan	Olgan bahosi
	Qayumova K.	2Am	Inf. va AT	4
	Isaev T.	3 Mat	Inf. va AT	4
	Alieva L.	1 Fiz	Inf. va AT	3
	Saidov B.	3 PXM	Inf. va AT	5
	Bozorova N.	4 Mat	Inf. va AT	4
	Boboev J.	2Am	Inf. va AT	5
	Zokirov S.	3 Am	Inf. va AT	5

Bu erda 3 3-yozuvni (Alieva L.),..., 1 1-yozuvni bildiradi. Ba'zan konkret faylni bir nechta kalit bo'yicha saralashga to'g'ri keladi.

Saralash 2 turga: ichki va tashqi saralashga bulinadi. Ichki saralashda operativ xotiradagi axborotlar qayta ishlanadi, tashqi saralashda tashqi xotiradagi axborotlar qayta ishlanadi. Optimallashtirish muammosi bu ikkala holda bir-biridan farq qiladi. Ichki saralashda kalitlarni taqqoslashlar va fayl yozuvlarining joyini o'zgartirishlar sonini kamaytirishga xarakat qilinadi. Tashqi saralashda mos algoritm effektivligining asosiy faktori disk qurilmalariga murojaatlar sonidir.

Saralash masalasini qo'yilishini quyidagicha yozish mumkin.

Faraz qilaylik, a_1, a_2, \dots, a_n , elementlar ketma-ketligi berilgan bo'lsin. U holda saralash algoritmi elementlarni massivga shunday joylashtiradiki, natijada ular qandaydir munosabatga nisbatan $f(a_{k1}) \leq f(a_{k2}) \leq \dots \leq f(a_{kn})$ tartibga ega bo'ladi. Odatda f tartiblash funksiyasi qandaydir maxsus qoida bilan hisoblanmasdan, balki elementni kalit qiymati bo'yicha massiv elementlari tartiblanadi.

Ma'lumotlarga qayta ishlov berilayotganda ma'lumotni informatsion maydonini hamda uni mashinda joylashishini (adresini) bilish zarur.

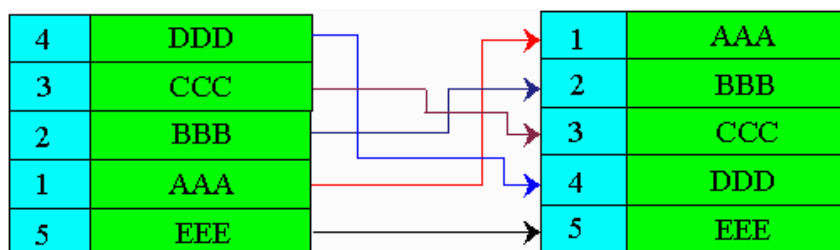
Saralashni ikkita turi mavjud: ichki va tashqi:

ichki saralash bu operativ xotiradagi saralash;

tashqi saralash – tashqi xotirada saralash.

Saralash bu ma'lumotlarni kalitlari bo'yicha xotirada regulyar ko'rinishda

joylashtirishdir. Regulyarlik deganda ma'lumotlar kalit qiymatlari bo'yicha massivda boshidan oxirigacha o'sishi yoki kamayishi tushiniladi.



Agar saralanayotgan yozuvlar xotirada katta xajmni egallasa, u holda ularni almashtirishlar katta sarf (vaqt va xotira ma'nosida) talab qiladi. Ushbu sarfni kamaytishi maqsadida, saralash kalitlar adresi jadvalida amalga oshiriladi. Bunda faqatgina ma'lumot ko'rsatkichlari almashtirilib, massiv o'z joyida qoladi. YUqoridagi usul adreslar jadvalini saralash usuli deyiladi.

Saralanayotganda bir hil kalitlar uchrashi mumkin, bu holda saralanagandan keyin bir hil kalitlilar boshlang'ich tartibda qanday joylashgan bo'lsa, ushbu tartibda qoldirilishi maqsadga muvofiq bo'ladi (Bir hil kalitlilar o'zlariga nisbatan). Bunday usulga turg'un saralash deyiladi.

Saralash samaradorligini bir necha mezonlar bo'yicha baholash mumkin:

saralashga ketgan vaqt;

saralash uchun talab qilingan operativ xotira;

dasturni ishlab chiqishga ketgan vaqt.

Birinchi mezonni qarab chiqaylik. Saralash bajarilganda taqqoslashlar yoki almashtirishlar soni hisoblash mumkin.

Faraz qilaylik, $N = 0,01n^2 + 10n$ – taqqoslashlar soni. Agar $n < 1000$ bo'lsa, u holda ikkinchi qo'shiluvchi katta, aks holda ya'ni, $n > 1000$ bo'lsa, birinchi qo'shiluvchi katta bo'ladi.

Demak, kichkina n larda taqqoslashlar soni n ga teng bo'ladi, katta n larda esa n^2 ga teng bo'ladi.

Saralashda taqqoslashlar soni quyidagi oraliqlarda bo'ladi:

$O(n \log n)$ dan $O(n^2)$ gacha; $O(n)$ – ideal holatda.

Saralashni quyidagicha usullari bor:

qat'iy (to'g'ridan-to'g'ri) usullar;

yaxshilangan usullar.

Qat'iy usullar:

1. to'g'ridan-to'g'ri qo'shish usuli;
2. to'g'ridan-to'g'ri tanlash usuli;
3. to'g'ridan-to'g'ri almashtirish usuli.

YUqorida keltirilgan uchala usulda ham almashtirishlar soni deyarli bir hil bo'ladi.

To'g'ridan-to'g'ri qo'shish usuli bilan saralash

Bunday usul karta o'yinida keng qo'llaniladi. Elementlar (kartlar) xayolan "tayyor" $a(1), \dots, a(i-1)$ va boshlang'ich ketma-ketliklarga bo'linadi. Har bir qadamda ($i=2$ dan boshlanib, har bir qadamda bir birlikka oshirib boriladi) boshlang'ich ketma-ketlikdan i -chi element ajratib olinib tayyor ketma-ketlikning kerakli joyiga qo'shiladi.

Taklif qilinayotgan usulni quyidagi misolda ko'rib chiqamiz.

Faraz qilaylik, kalit qiymati 4, 5, 3, 8, 1, 7 bo'lgan elementlar berilgan bo'lsin.

$i = 2$	4	5	3	8	1	7
$i = 3$	4	3	5	8	1	7
	3	4	5	8	1	7
$i = 4$	3	4	5	8	1	7
$i = 5$	1	3	4	5	8	7
$i = 6$	1	3	4	5	7	8

Kerakli joyni qidirish jarayonini quyidagi tartibda olib borish qulay bo'ladi. Taqqoslashlar amalga oshirish mobaynida, navbatdagi $a(j)$ element bilan solishtiriladi, keyin esa x bo'sh joyga qo'yiladi yoki $a(j)$ o'nga suriladi va jarayon chapga "ketadi". SHuni e'tiborga olish lozimki, saralash jarayoni quyidagi shartlarni birortasi bajarilganda yakunlanadi:

1. x elementi kalitidan kichik kalitli $a(j)$ element topildi.
2. tayyor ketma-ketlikning chap tomoni oxiriga etib borildi.

Taklif etilayotgan usul algoritmi quyidagicha bo'ladi:

Procedure StraightInsertion

```

Var
  i,j:index; x:item;
begin
  for i:=2 to n do
    x:=a[i]; a[0]:=x; j:=1;
    while x<a[j-1] do a[j]:=a[j-1]; j:=j-1; end;
    a[j]:=x
  end
end StraightInsertion

```

algoritm samaradorligi

Faraz qilaylik, taqqoslashlar soni S , o'rinlashtirishlar soni M bo'lsin. Agar massiv elementlari kamayish tartibida bo'lsa, u holda taqqoslashlar soni eng katta bo'lib, u $C_{\max} = \frac{n(n-1)}{2}$ ga teng bo'ladi, ya'ni $O(n^2)$. O'rinlashtirishlar soni esa $M_{\max} = C_{\max} + 3(n-1)$ ga teng bo'ladi, ya'ni $O(n^2)$. Agar berilgan massiv o'sish tartibida saralangan bo'lsa, u holda taqqoslashlar va o'rinlashtirishlar soni eng kichik bo'ladi, ya'ni $C_{\min} = n-1$, $M_{\min} = 3(n-1)$.

To'g'ridan-to'g'ri tanlash usuli bilan saralash

Faraz qilaylik, a_1, a_2, \dots, a_n elementlar ketma-ketligi berilgan bo'lsin.

Mazkur usul quyidagi tamoyillarga asoslangan:

1. Berilgan elementlar ichidan eng kichik kalitga ega element tanlanadi.
2. Ushbu element boshlang'ich ketma-ketlikdagi birinchi element a_1 bilan o'rin almashadi.
3. Undan keyin ushbu jarayon qolgan $n-1$ ta element, $n-2$ ta element va xokazo, toki bitta eng "katta" element qolguncha davom ettiriladi.

Taklif qilinayotgan usul algoritmi quyidagicha bo'ladi:

Paskal tilidagi dasturi:

Procedure StraightSelection

Var

```

        i,j,k: index; x:item;
begin
    for i:=1 to n-1 do
        k:=I; x:=a[i];
        for j:=i+1 to n do
            if a[j]<x then k:=j; x:=a[k]
        end;
    end;
    a[k]:=a[i];
    a[i]:=x
end;
end StraightSelection

```

Algoritm samaradorligi:

$$\text{Taqqoslashlar soni } M = \frac{n}{2}(n-1) = \frac{n^2 - n}{2}.$$

Almashtirishlar soni $C_{\min} = 3(n-1)$, $C_{\max} = 3(n-1)\frac{n}{2}$
(n^2 tartib).

Ushbu usul bo'yicha saralash bajarilsa, eng yomon xolda taqqoslashlar va almashtirishlar soni tartibi n^2 bo'ladi.

To'g'ridan-to'g'ri almashtirish usuli bilan saralash (pufaksimona)

Ushbu usulni g'oyasi quyidagicha: $n-1$ marta massivda quyidan yuqoriga qarab yurib kalitlar jufti-jufti bilan taqqoslanadi. Agar pastki kalit qiymati yuqoridagi jufti kalitidan kichik bo'lsa, u holda ular o'rnini almashtiriladi.

O'tish raqami	1	2	3	4	5
	4	1	1	1	1
	3	4	2	2	2
	7	3	4	3	3
	2	7	3	4	4
	1	2	7	5	5
	6	5	5	6	6
	5	6	6	7	7

Paskal tilidagi dasturi:

```

for i := 2 to n do
  for j := n downto i do
    if a[j - 1] > a[j] then
      begin
        x := a[j - 1];
        a[j - 1] := a[j];
        a[j] := x;
      end;

```

Bizning xolatda bitta o'tish "bekor" bo'ldi. Elementlarni ortiqcha o'rinashtirmaslik uchun bayroqcha kiritish mumkin.

Pufaksimom usulni yaxshilangan usuli bu sheyker saralash usuli bo'lib, har bir o'tishdan keyin tsikl ichida yo'nalish o'zgartiriladi.

Algoritm samaradorligi:

$$\text{taqqoslashlar soni } M = \frac{n}{2} \cdot \frac{n}{2} = \frac{n^2}{4},$$

$$\text{almashtirishlar soni } C_{\max} = 3 \frac{n^2}{4}.$$

Saralashning yaxshilangan usullari

Quicksort – tez saralash usuli

K.Xoorning tez saralash algoritmi bo'lishli saralash deb ataldi. Ushbu algoritm boshqa saralash usullariga nisbatan vaqt bo'yicha yaxshi natijalar ko'rsatadi. Tez saralash usulining mohiyati quyidagidan iborat:

S (k) (k=1,2,...,n)- bir o'lchovli massiv berilgan bo'lsin. x S (k) dan olingan

qandaydir element bo'lsin. Bunda S shunday ikkita S_1 va S_2 ($S_1 \cup S_2 = S$) kesishmaydigan bo'sh emas qismlarga bo'linadiki, S_1 dagi elementlar x dan katta bo'lmasin, S_2 dagi elementlar esa x dan kichik bo'lmasin:

6,23,17,8, 14,25,6,3,30,7

$X=14$; S ni qandaydir $a > x$ element uchraguncha tekshiramiz: $a=23$; Keyin S ni qandaydir $b < x$ element topilguncha o'ngdan chapga tekshiramiz: $b=7$; a va b larning o'rinlarini almashtiramiz. Jarayonni davom ettirib, quyidagi ketma-ketlikka ega bo'lamiz:

6,7,3,8,6 □ 14, 25,17,30,23.

Shunday qilib, S_1 va S_2 bo'laklar ham xuddi yuqoridagi kabi saralanadi. Jarayon har bir bo'lakda bittadan element qolguniga qadar davom ettiriladi. Natijada saralangan massivga ega bo'lamiz.

```
procedure Sort (L, R: integer);
```

```
begin
```

```
  i := L;
```

```
  j := r;
```

```
  x := a[(L + r) div 2];
```

```
  repeat
```

```
    while a[i] < x do
```

```
      i := i + 1;
```

```
    while a[j] > x do
```

```
      j := j - 1;
```

```
    if i <= j then
```

```
      begin
```

```
        y := a[i];
```

```
        a[i] := a[j];
```

```
        a[j] := y;
```

```
        i := i + 1;
```

```
        j := j - 1
```

```
      end;
```

```

until i > j;
if L < j then sort (L, j);
if i < r then sort (i, r);
end;

```

```

procedure QuickSort;
begin
  sort (1, n);
end;

```

Algoritm samaradorlig:

$O(n \log n)$ – eng samarali usul.

Nazorat savollari

1. Saralash deganda nimani tushunasiz?
2. Saralashning asosiy usullarini aytib bering.
3. Saralashning qaysi usulari qat'iy usulga tegishli?
4. Saralashning yaxshilangan usullarini aytib bering.
5. Qanday saralash turg'un deyiladi?
6. To'g'ridan-to'g'ri qo'shish usuli g'oyasi nimadan iborat?

7-Ma'ruza: Saralash usullarini tanlashda hisobga olinadigan omillar.

Reja

1. To'g'ridan to'g'ri almashtirish orqali saralash.
2. Saralashning yaxshilangan usullari.

SHell saralashi (qisqarib boruvchi qadamlar orqali saralash)

To'g'ri qo'shish usulini 1959 yilda D. SHell tomonidan mukammallashtirish taklif qilingan. Quyidagi chizmada ushbu usul tasvirlangan:

Saralas	4	5	2	2	4	8		7
h								
Keyin								

to'rtlik	4	8		2	4	5	2	7
Ikkilik		8	2	2	4	5	4	7
Yakkalik								
ik		2	8	2	4	5	7	4

Boshida bir biridan 4 qadamda joylashgan elementlar o'zaro guruhlanib saralash amalga oshiriladi. Bunday jarayon to'rtlik saralash deb ataladi. Birinchi o'tishdan keyin elementlar qayta guruhlanib, endi har ikki qadamdagi elementlar taqqoslanadi. Bu esa ikkilik saralash deb nomlanadi. Va nihoyat, uchinchi o'tishda oddiy yoki yakkalik saralashi amalga oshiriladi.

Bir qarashda mazkur usul bilan saralash amalga oshirilganda saralash jarayoni kamayish o'rniga ortib boradigandek tuyulsada, elementlarni o'rin almashtirishlar nisbatan kam amalga oshiriladi.

Ko'rinib turibdiki, bu usul natijasida tartiblangan massiv hosil bo'lib, har bir o'tishdan keyin saralashlar kamayib boradi. Eng yomon holatda oxirgi ishni yakkalik saralash amalga oshiradi.

Bar'er usulidan foydalanilganda har bir saralash o'zining bar'eriga ega bo'lishi lozim hamda dastur uning joyini aniqlashi uchun uni iloji boricha osonlashtirish lozim. SHuning uchun massivni $[-h1..N]$ gacha kengaytirish lozim bo'ladi.

$h[1..t]$ – qadamlar o'lchami massivi

$a[1..n]$ - saralanayotgan massiv

k – saralash qadami

x – qo'shilayotgan element qiymati

Subroutine ShellSort

const $t = 3;$

$h(1) = 5;$

$h(2) = 3;$

$h(3) = 1;$

var

h : array $[1..t]$ of integer;


```

a: array [1..n] of integer;
k, x, m, t, i, j: integer;
begin
  for m = 1 to t do
    begin
      k:= h(m);
      for i = k + 1 to n do
        begin
          x:= a(i);
          for j = i-k downto k do
            begin
              if x < a(j) then
                a(j+k):= a(j);
              else
                goto 1;
            end ;
          end;
        end;
      end;
    1:    a(j+1) = x ;
          end;
        end .

```

Umuman olganda, qanday qadamlar tanlanganda eng yaxshi natija olinishi isbotlanmagan bo'lsada, lekin bu qadamlar biri ikkinchisini ko'paytuvchilari bo'lmasligi lozimligi aniqlangan.

D. Knut qadamlarni quyidagicha ketma-ketligini taklif qilgan (teskari tartibda): 1,3,7,15,31,...,ya'ni: $h_{m-1}=2h_m+1$, $h_t=1$, $t = \lceil \log_2 n \rceil - 1$. Agar qadamlar ushbu ko'rinishda aniqlansa, algoritm samaradorligi tartibi $O(n \log n)$.

Daraxt usulida saralash

Saralashning barcha usullari S massiv elementlarini ko'rib chiqish va ular ustida qandaydir amallar bajarishdan iboratdir. Bunday algoritmlardan biri

saralanayotgan S massivni binar D daraxt ko'rinishida ifodalashdir. Quyida uning sxematik tasvirini keltiramiz:

```

142  83
14   55                                     46   97
128  39   1710   111   312

```

Bunda S massiv: 9 14 8 1 5 4 9 12 3 17 1 3 elementlaridir;

Bu erda $8 \leq n \leq 16$; 1 dan boshlangan natural sonlar bilan yuqoridan pastga va chapdan ungra qarab D daraxtning barcha uchlari nomerlab chiqilgan. Ushbu nomerlar adreslar rolini bajaradi. Binar D daraxtda bitta ildiz tugun bo'lib, uning ajdodi bo'lmaydi. Tugunning adresi -1; ixtiyoriy boshqa tugunlar bitta ajdodga va bitta yoki ikkita avlodga ega bo'ladi. Ba'zi hollarda daraxtlar ikki o'lchovli massivlar ko'rinishida har bir tugunning ajdod va avlodlari uchun adreslari oshkor tarzda ifodalanadi $k(k=1,n)$. Ammo real holatda bu adreslarni saqlash emas, r nomer bo'yicha hisoblash osonroqdir:

a) ajdodlar uchun: $x(k) = k/2, k=1,2,\dots,p;$

$2^*k, k=1,2,\dots,n/2$

b) chapdagi avlod uchun $u(k)=0, k > n/2$

$2^*k+1, k=1,2,\dots,(n-1)/2$

v) o'ngdagi avlod uchun $z(k)=0, k > (n-1)/2$

Daraxtdagi ixtiyoriy tugun boshqa daraxt uchun ildiz vazifasini bajarishi mumkin.

12.4.Piramidali saralash.

Piramidali saralash Dj.Uilyams tomonidan taklif etilgan va R.Floyt tomonidan rivojlantirilgan. Bunda S massiv D binar daraxt ko'rinishida ifodalanadi va qo'shimcha xotira talab etmaydi. Algoritmning bajarilish murakkabligi $O(n \log_2 n)$ ga teng.

$S(1), S(2), \dots, S(n)$ (1)

massiv berilgan bo'lsin. (5) elementlarning

$S(p), S(p+1), \dots, S(q)$ ($1 < p < q < n$) (2)

Ko'rinishdagi ketma-ketlik piramida deb ataladi, qachonki quyidagi shartlardan biri bajarilsa:

- 1) $2p > q$
- 2) $lp = q, S(p) > S(q)$
- 3) $2p < q, S(j) > S(2j)$ ($p < j < q/2$)

Ta'rifdan quyidagilar kelib chiqadi:

Ixtiyoriy (5) ketma-ketlik uchun $S(n/2+1), S(n/2+2), \dots, S(n)$ ketma-ketlik piramida bo'lib hisoblanadi;

Agar (1) ketma-ketlik piramida bo'lsa, u holda $S(1) > \max S(j)$ (3)

Agar (1) ketma-ketlik piramida bulib, binar D daraxt kurinishida berilgan bo'lsa, D dagi ixtiyoriy tugunning qiymati uning chap va o'ng avlodlari qiymatidan kichik bo'lmaydi.

2-Misol. 90, 70, 11, 8, 3, 9, 7, 5, 6, 1, 2 ketma-ketlik berilgan va u piramidadir:

90				
70	11			
8	3	9	7	
5	6		1	2

Piramidali saralash ikki etapdan iborat buladi:

1-etap. Piramidani qurish. (1) ketma-ketlikda

$S(n/2+1), S(n/2+2), \dots, S(n)$ (4)

Piramidadir. (4) ketma-ketlikka (1) dan qolgan elementlarni qo'shamiz.

$S(j+1), S(j+2), \dots, S(n)$ piramida bo'lsin. CHapdan $S(j)$ elementni qo'shib,

$S(a), S(j+1), S(j+2), \dots, S(n)$ (5)

(5) ni yana piramidaga aylantiraymiz, ya'ni $S(j)$ va uning ikkita avlodi $S(2j)$ va $S(2j+1)$ lar tekshiriladi. Bunda agar $S(j)$ avlodlaridan kichik bo'lmasa hisoblashlar to'xtatiladi, chunki (5) piramida bo'lib hisoblanadi. Aks holda $S(j)$ va $\max(S(2j), S(2j+1))$ qiymatlarni almashtiramiz va h.k.z. Oxirida (1) piramidga aylanadi va (3)

bajariladi. Olingan S piramidani joriy deb e'lon qilamiz va 2-etapga o'tamiz.

2-etap. Joriy S piramidada 1-element qolganlaridan kichik emas. S ning chekka elementlari qiymatlarini o'zaro almashtirib, S ni uning tomondan bittaga qisqartiramiz. Hosil bo'lgan ketma-ketlik piramida bo'lmasligi ham mumkin. S(1) element uchun 1-etapdagi jarayonni qo'llab, o'zgartirilgan S ketma-ketlik yana piramidaga aylantiriladi. 2-etapni p-1 marata bajarib, S ni o'smaslik tartibida saralab olamiz.

Ushbu saralash usulini konkret misolda ko'rib o'tamiz.

4-misol.

23, 77, 12, 7, 44, 82, 16, 53 ketma-ketlik uchun piramidali saralash o'tkazamiz. Bunda algoritmi bajarilish jarayonidagi S ketma-ketlikning joriy elementlari yozib olinsin.

Quyida S ketma-ketlikning algoritmi bajarilishining har bir 1 va 2 etap realizatsiyasidagi qiymatlari ko'rsatilgan.

Piramidani qurish

23	77	12	7	44	82	16	53
23	77	12	53	44	82	16	7
23	77	82	53	44	12	16	7
23	77	82	53	44	12	16	7
82	77	23	53	44	12	16	7

Piramidani saralash

7	77	23	53	44	12	17	82
16	53	23	7	44	12	77	82
12	44	23	7	16	53	77	82
12	16	23	7	44	53	77	82
7	16	12	23	44	53	77	82
12	7	16	23	44	53	77	82
7	12	16	23	44	53	77	82

Piramidali saralash usulining analizi shuni ko'rsatadiki, uning bajarilishi uchun $3n \log_2 n$ tadan ko'p bo'lmagan elementar operatsiya bajarilishi talab etiladi.

Nazorat savollari

1. To'g'ridan-to'g'ri tanlash usuli g'oyasi nimadan iborat?
2. To'g'ridan-to'g'ri almashtirish usuli g'oyasi nimadan iborat?
3. Yuqoridagi usullarning bir biridan farqini aytib bering.
4. Qaysi saralash usuli eng samarali bo'lib hisoblanadi?
5. Shell usuli qaysi asosiy saralash usuliga tegishli?

8-ma'ruza:Ma'lumotlarni izlash algoritmlari.Ketma-ket izlash. Izlashning tezashtiririlgan usullari.

Reja:

1. Ketma-ket qidiruv
2. Indeksli ketma-ket qidiruv
3. Ketma-ket qidiruvni samaradorligi
4. Indeksli ketma-ket qidiruvni samaradorligi
5. Qidiruvni mukammallashtirish usullari
6. Topilgan elementni ro'yxat boshiga qo'yish orqali jadvalni qayta tartiblash
7. Transpozitsiya usuli
8. Mukammal qidiruv daraxti
9. Binar qidiruv (teng ikkiga bo'lish usuli)

O'quv maqsadi:

Ta'limiy: Talabalarda qidiruv usullari to'g'risida bilim va ko'nikmalarni shakllantirish.

Tarbiyaviy: Qidiruv orqali bajariladigan ishlar, ularning ahamiyatini tushuntirish orqali talabalarning qiziqishini oshirish, ulardan foydalanish madaniyatini shakllantirish, ilmga intiluvchanlik ruhida tarbiyalash;

Rivojlantiruvchi: Talabalarning "Ma'lumotlarni qidiruv usullari" mavzusidan

olgan bilimlarini mutaxassislik sohasidagi ishlarida qo'llash orqali rivojlantirish.

EXMda ma'lumotlarni qayta ishlashda qidiruv asosiy amallardan biri bo'lib hisoblanadi. Uning vazifasi berilgan argument bo'yicha massiv ma'lumotlari ichidan mazkur argumentga mos ma'lumotlarni topishdan iborat.

Ixtiyoriy ma'lumotlar majmuasi jadval yoki fayl deb ataladi. Ixtiyoriy ma'lumot (yoki tuzilma elementi) boshqa ma'lumotdan biror bir belgisi orqali farq qiladi. Mazkur belgi kalit deb ataladi. Kalit noyob bo'lishi, ya'ni mazkur kalitga ega ma'lumot jadvalda yagona bo'lishi mumkin. Bunday noyob kalitga boshlang'ich (birinchi) kalit deyiladi. Ikkinchi kalit bir jadvalda takrorlansada u orqali ham qidiruvni amalga oshirish mumkin. Ma'lumotlar kalitini bir joyga yig'ish (boshqa jadvalga) yoki yozuv sifatida ifodalab bitta maydonga kalitlarni yozish mumkin. Agar kalitlar ma'lumotlar jadvalidan ajratib olinib alohida fayl sifatida saqlansa, u holda bunday kalitlar tashqi kalitlar deyiladi. Aks holda, ya'ni yozuvning bir maydoni sifatida jadvalda saqlansa ichki kalit deyiladi.

Kalitni berilgan argument bilan mosligini aniqlovchi algoritmgga berilgan argument bo'yicha qidiruv deb ataladi. Qidiruv algoritmi vazifasi kerakli ma'lumotni jadvalda topish yoki yo'qligi aniqlashdan iboratdir. Agar kerakli ma'lumot yo'q bo'lsa, u holda ikkita ishni amalga oshirish mumkin:

ma'lumot yo'qligini indikatsiya (belgilash) qilish
jadvalga ma'lumotni qo'yish.

Faraz qilaylik, k – kalitlar massivi. Har bir $k(i)$ uchun $r(i)$ – ma'lumot mavjud. Key – qidiruv argumenti. Unga rec - informatsion yozuv mos qo'yiladi. Jadvaldagi ma'lumotlarning tuzilmasiga qarab qidiruvni bir necha turlari mavjud.

1. Ketma-ket qidiruv

	i	k	r	
indeks	1	8	...	Information maydon
	2	136	...	
	3	4	...	
	
kalitlar	n-1	17	...	
	n	234	...	

Paskal tilida dastur quyidagicha bo'ladi:

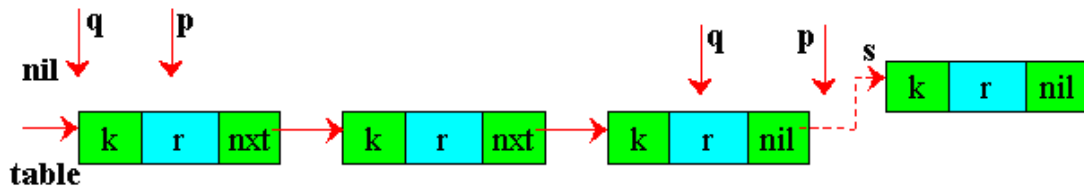
```
for i:=1 to n do
  if k[i] = key then
    begin
      search = i;
      exit;
    end;
search = 0;
exit;
```

Massivda ketma-ket qidiruv algoritmi samaradorligini bajarilgan taqqoslashlar soni M bilan aniqlash mumkin. $M_{min} = 1$, $M_{max} = n$. Agar ma'lumotlar massiv yacheykasida bir xil extimollik bilan taqsimlangan bo'lsa, u holda $M_{sr} \approx (n + 1)/2$ bo'ladi.

Agar kerakli element jadvalda bo'lmasa va mazkur elementni jadvalga qo'shish lozim bo'lsa, u holda yuqoridagi dasturdagi oxirgi ikkita operator quyidagiga almashtiriladi.

```
Paskalda
n:=n+1;
k[n]:=key;
r[n]:=rec;
search:=n;
exit;
```

Agar ma'lumotlar jadvali bir bog'lamli ro'yxat ko'rinishida berilgan bo'lsa, u holda ketma-ket qidiruv ro'yxatda amalga oshiriladi.



Algoritmlar varianti:

Paskalda:

q:=nil;

p:=table;

while (p <> nil) do

begin

if p^{.k} = key then

begin

search = p;

exit;

end;

q := p;

p := p^{.nxt};

end;

New(s);

s^{.k}:=key;

s^{.r}:=rec;

s^{.nxt}:= nil;

if q = nil then table = s

else q^{.nxt} = s;

search:= s;

exit

Ro'yxatli tuzilmaning afzalligi shundan iboratki, ro'yxatga elementni qo'shish yoki o'chirish tez amalga oshadi, bunda qo'shish yoki o'chirish element soniga bog'liq bo'lmaydi, massivda esa elementni qo'shish yoki o'chirish taxminan barcha elementlarni yarimini siljitishni talab qiladi. Ro'yxatda qidiruvni samaradorligi taxminan massivniki bilan bir hil bo'ladi.

Umuman olganda ketma-ket qidiruv samaradorligini oshirish mumkin.

Faraz qilaylik, kun davomida ma'lumotlar yig'ilib, kechqurun ular qayta ishlansin. Ma'lumotlar to'plangandan keyin ular saralanadi.

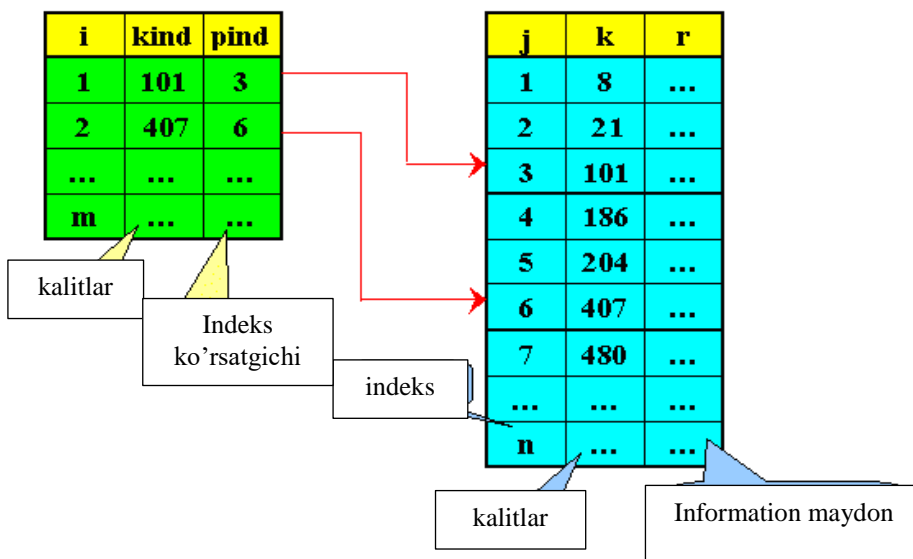
2. Indeksli ketma-ket qidiruv

Mazkur ko'rinishdagi qidiruv amalga oshirilayotganda ikkita jadval tashkil qilinadi: o'z kalitiga ega ma'lumotlar jadvali (o'sish tartibida tartiblangan) va indekslar jadvali, bu xam ma'lumotlar kalitidan iborat-u, lekin bu kalitlar asosiy jadvaldan aniq bir interval orqali olingan. (2-chizma).

Boshida berilgan argument bo'yicha ketma-ket qidiruv indekslar jadvalida amalga oshiriladi. Qachonki, biz berilgan kalitdan kichik kalitni aniqlaganimizda, asosiy jadvalda qidiruvni quyi chegarasini o'rnatamiz - low, keyin esa yuqori chegarani - hi, ya'ni ($kind > key$).

Masalan, $key = 101$.

Qidiruv to'la jadval bo'yicha emas, balki low dan hi gacha davom etadi.



2-CHizma.

Dasturga misol:

Paskal':

i:=1;

while (i <= m) and (kind[i] <= key) do

 i=i+1;

if i = 1 then low:=1 else low:=pind[i-1];

if i = m+1 then hi:=n else hi:=pind[i]-1;

for j:=low to hi do

 if key = k[j] then

 begin

 search:=j;

 exit;

 end;

search:=0;

exit;

3. Ketma-ket qidiruvni samaradorligi

Ixtiyoriy qidiruvning samaradorligi jadvaldagi ma'lumotlarning kalitlari bilan solishtirish soni – S bilan baxolanishi mumkin. Agar taqqoslashlar (solishtirish) soni qancha kichik bo'lsa, qidiruv algoritmi samaradorligi shuncha yaxshi bo'ladi.

Massivda ketma-ket qidiruvning samaradorligi quyidagicha bo'ladi:

$$C = 1 \div n, C = (n + 1)/2.$$

Umuman olganda ro'yxatda xam samaradorlik yuqoridagi kabi bo'ladi. Garchi massivda xam bog'langan ro'yxatda xam qidiruv samaradorligi bir xil bo'lsada, ma'lumotlarni massiv va ro'yxat ko'rinishda tasvirlashning o'ziga xos kamchilik va afzalliklari mavjud. Qidiruvning maqsadi - quyidagi jarayonlarni bajarilishidan iborat:

Topilgan yozuvni o'qish.

Qidirilayotgan yozuv topilmasa, uni jadvalga qo'yish.

Topilgan yozuvni o'chirish.

Birinchi jarayon (qidiruvning o'zi) massiv uchun ham ro'yxat uchun ham bir xil bo'ladi. Ikkinchi va uchinchi jarayonda esa qidiruv ro'yxatli tuzilmada samaraliqroq bo'ladi (sababi massivda elementlarni siljitish lozim).

Agar k massivda elementlarni siljitishlar soni bo'lsa, u xolda $k = (n + 1)/2$ bo'ladi.

4. Indeksli ketma-ket qidiruvni samaradorligi

Agar bo'lishi mumkin barcha xolatlar teng extimolli deb olinsa, u holda qidiruv samaradorligini quyidagicha xisoblash mumkin:

Belgilashlar kiritib olamiz: m – indeks o'lchovi; $m = n / p$; p – qadam o'lchovi

$$Q = (m+1)/2 + (p+1)/2 = (n/p+1)/2 + (p+1)/2 = n/2p + p/2 + 1 \quad (*)$$

Q ni p bo'yicha differentsiallab uni nolga tenglashtiramiz:

$$dQ/dp = (d/dp) (n/2p + p/2 + 1) = -n/2p^2 + 1/2 = 0$$

Bu erdan

$$p^2 = n; \quad p_{onm} = \sqrt{n}$$

(*) ifodada r o'rniga ropt ni qo'yib quyidagi taqqoslashlar sonini olamiz:

$$Q = \sqrt{n} + 1$$

Demak, indeksli ketma-ket qidiruvni samaradorligi tartibi $O(\sqrt{n})$ bo'ladi.

Nazorat savollari:

1. Qidiruv vazifasi nimadan iborat?
2. Noyob kalit deganda nimani tushunasiz?
3. Ro'yxatda berilgan kalitli element yo'q bo'lganda qaysi amal bajariladi?
4. Ketma-ket qidiruv va indeksli ketma-ket qidiruvlarning farqi nimadan iborat?
5. Ulardan qaysi biri samaraliroq va nima sababdan?
6. Jadvalni qayta tartiblashning qanday usullarini bilasiz?

9-ma'ruza:Ma'lumotlardan bevosita erkin foydalanadigan izlash usuli.

1. Qidiruvni mukammallashtirish usullari

Umuman olganda, jadvalda xar bir elementni qidirish extimolligini qandaydir bir qiymat bilan izohlash mumkin. Faraz qilaylik jadvalda qidirilayotgan element mavjud. U holda qidiruv amalga oshirilayotgan barcha jadvalni diskret xolatga ega tizim sifatida qarash mumkin xamda unda qidirilayotgan elementni topish extimolligi – bu tizim i-chi xolati extimolligi $p(i)$ deb olish mumkin.

$$\sum_{i=1}^n p(i) = 1$$

Jadvalni diskret tizim sifatida qaraganimizda, undagi taqqoslashlar soni diskret tasodifiy miqdorlar qiymatlarini matematik kutilmasini ifodalaydi.

$$Z=Q=1p(1)+2p(2)+3p(3)+\dots+np(n)$$

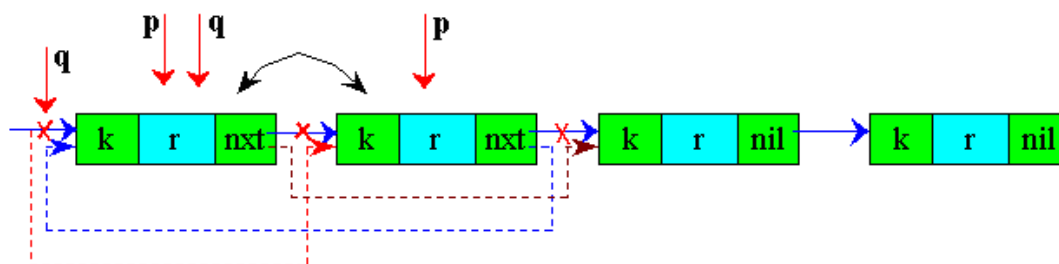
Iloji boricha $p(1) \geq p(2) \geq p(3) \geq \dots \geq p(n)$ bo'lsa, maqsadga muvofiq bo'ladi.

Bu shart taqqoslashlar sonini kamaytirib, samaradorlikni oshiradi. Sababi, ketma-ket qidiruv birinchi elementdan boshlanganligi uchun eng ko'p murojaat qilinadigan elementni birinchiga qo'yish lozim.

Qidiruv jadvalini qayta tartiblashni eng ko'p ishlatiladigan ikkita usuli mavjud. Ularni bir bog'lamli ro'yxatlar misolida ko'rib chiqamiz.

2. Topilgan elementni ro'yxat boshiga qo'yish orqali jadvalni qayta tartiblash

Mazkur usulni mag'zi shundan iboratki, berilgan kalitga teng kalitli element ro'yxatda birinchi element deb o'zlashtiriladi, qolganlari esa suriladi.



Keltirilgan algoritm ro'yxat uchun ham massiv uchun xam o'rinli. Biroq bu

algoritm massiv uchun tavsiya qilinmaydi, sababi elementlarni o'rinlashtirishga ko'rsatkichlarni o'rinlashtirishdan ko'ra ancha ko'p vaqt talab qiladi.

Ro'yxatni qayta tartiblash algoritmi:

Paskal':

q:=nil;

p:=table;

while (p <> nil) do

begin

if key = p^.k then

begin

if q = nil

then 'o'rinlashtirish shart emas'

search := p;

exit;

end;

q^.nxt := p^.nxt;

p^.nxt := table;

table := p;

exit;

end;

q := p;

p := p^.nxt;

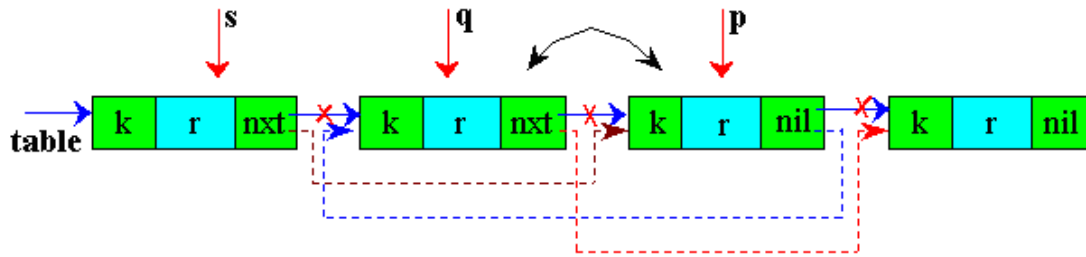
end;

search := nil;

exit;

3. Transpozitsiya usuli

Ushbu usulda topilgan element ro'xatda bitta oldingi element bilan o'rin almashtiriladi. Agarda mazkur elementga ko'p murojaat qilinsa, bittadan oldinga surilib borib natijada ro'yxat boshida bo'ladi.



CHizma. Qo'shni elementlarni o'rnini almashtirish

r – ishchi ko'rsatkich

q – yordamchi ko'rsatkich, r dan bitta qadam orqada bo'ladi

s - yordamchi ko'rsatkich, q dan ikkita qadam orqada bo'ladi

Transpozitsiya usuli algoritmi:

Paskal':

s:=nil;

q:=nil;

p:=table;

while (p <> nil) do

begin

if key = p^.k then

 'transponerlaymiz

 begin

 if q = nil then

 begin

 'o'rinlashtirilmaydi

 search:=p;

 exit;

 end;

 q^.nxt:=p^.nxt;

 p^.nxt:=q;

 if s = nil then

```

    table := p;
else
    begin
        s^.nxt := p;
    end;
search:=p;
exit;
end; end;
search:=nil; exit;

```

Ushbu usul nafaqat ro'yxatda, balki massivda xam qulay (sababi faqatgina ikkita yonma-yon turgan element o'rin almashtiriladi).

4. Mukammal qidiruv daraxti

Agar ajratib olingan elementlar qandaydir o'zgarmas to'plamni tashkil qilishsa, kelgusi qidiruvlar samaraliroq bo'lishi uchun ularni binar daraxt ko'rinishida ifodalash maqsadga muvofiq bo'lishi mumkin.

Quyida keltirilgan daraxtlarda binar qidiruvni ko'rib chiqaylik (a)va b) chizma). Ikkala daraxt xam uchtadan elementga ega - k_1, k_2, k_3 bo'lib bu erda $k_1 < k_2 < k_3$. k_3 elementni qidirish a) chizmada ikkita taqqoslashni talab qilsa, b) chizmada esa bitta.

Biror bir yozuvni ajratib olish uchun zarur bo'lgan kalitlarni taqqoslashlar soni binar qidiruv daraxtidagi ushbu yozuv bosqichiga birni qo'shganiga teng.

Faraz qilaylik:

qidiruv argumenti $key = k_1$ bo'lishi extimolligi - p_1 ,

qidiruv argumenti $key = k_2$ bo'lishi extimolligi - p_3 ,

qidiruv argumenti $key = k_3$ bo'lishi extimolligi - p_3 ,

$key < k_1$ bo'lishi extimolligi - q_0 ,

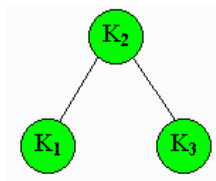
$k_2 > key > k_1$ bo'lishi extimolligi - q_1 ,

$k_3 > key > k_2$ bo'lishi extimolligi - q_2 ,

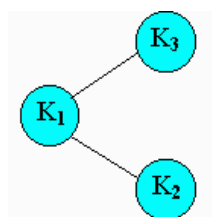
key > k₃ bo'lishi extimolligi – q₃,

C1 - a) chizmadagi taqqoslashlar soni,

C2 - b) chizmadagi taqqoslashlar soni.



Chizma a)



Chizma b)

U holda $r_1 + r_2 + r_3 + q_0 + q_1 + q_2 + q_3 = 1$

Biror bir qidiruvda kutilayotgan taqqoslashlar soni quyidagicha bo'ladi:

$$C1 = 2 \cdot r_1 + 1 \cdot r_2 + 2 \cdot r_3 + 2 \cdot q_0 + 2 \cdot q_1 + 2 \cdot q_2 + 2 \cdot q_3$$

$$C2 = 2 \cdot r_1 + 3 \cdot r_2 + 1 \cdot r_3 + 2 \cdot q_0 + 3 \cdot q_1 + 3 \cdot q_2 + 1 \cdot q_3$$

Biror bir berilgan kalitlar va extimolliklar to'plamida kutilayotgan taqqoslashlar sonini minimallashtiruvchi binar qidiruv daraxti mukammal deyiladi. Garchi daraxt yaratish algoritmi ancha sarmashaqqat ish bo'lsada, biroq u yaratgan daraxtda qiduvni amalga oshirish ancha samarali bo'ladi. Afsuski, ko'pincha, qidiruv argumenti extimolligi oldindan aniq bo'lmaydi.

9. Binar qidiruv (teng ikkiga bo'lish usuli)

Faraz qilaylik, o'sish tartibida tartiblangan sonlar massivi berilgan bo'lsin. Ushbu usulni asosiy g'oyasi shundan iboratki, tasodifiy qandaydir AM element olinadi va u X qidiruv argumenti bilan taqqoslanadi. Agar $AM = X$ bo'lsa, u holda qidiruv yakunlanadi; agar $AM < X$ bo'lsa, u holda indeksleri M dan kichik yoki teng bo'lgan barcha elementlarni kelgusi qidiruvdan chiqarib yuboriladi. Xuddi shuningdek, agar $AM > X$ bo'lsa.

M ixtiyoriy tanlanganda ham taklif qilinayotgan algoritm korrekt ishlaydi. SHu sababali M ni shunday tanlash lozimki, tadqiq qilinayotgan algoritm samaraliroq natija bersin, ya'ni uni shunday tanlaylikki, iloji boricha kelgusi jarayonlarda ishtirok etuvchi elementlar soni kam bo'lsin. Agar biz o'rtacha elementni, ya'ni massiv o'rtasini tanlasak echim mukammal bo'ladi.

Quyidagi chizma ko'rinishida berilgan massivni qarab chiqaylik. Faraz qilaylik, bizdan kaliti 52 ga teng bo'lgan elementni topish talab qilinsin. Dastlabki taqqoslanadigan element 49 bo'ladi. $49 < 52$ bo'lgani uchun sababli, keyingi qidiruv 49 dan yuqorida turgan elementlar orasida amalga oshiriladi. Yangi xosil bo'lgan massiv o'rtasi 86. Agar berilgan kalit bilan ushbu kalitni taqqoslasak $86 > 52$ bo'ladi. Demak, navbatdagi qidiruvlar 86 bilan 49 orasidagi elementlar ichida amalga oshirilishi lozim. Keyingi qadamda ma'lum bo'ldiki, qaralayotgan elementlar o'rtasidagi element kaliti 52 ga teng. Shunday qilib, massivda berilgan kalitga teng bo'lgan elementni topdik.

Yuqoridagi algoritmnini amalga oshirish dasturi Paskal tilida quyidagicha bo'ladi:

Paskalda

```
low := 1;
hi := n;
while (low <= hi) do
  begin
    mid := (low + hi) div 2;
    if key = k[mid] then
      begin
        search := mid;
        exit;
      end;
    if key < k[mid]
      then hi := mid - 1
      else low := mid + 1;
    end;
  search := 0;
  exit
```

Agar key = 101 bo'lsa, kerakli yozuv 3 marta taqqoslashlarda aniqlanadi (ketma-ket qidiruvda taqqoslashlar soni 7 ta bo'lar edi).

Agar S – taqqoslashlar soni va n – jadvaldagi elementlar soni bo'lsa, u holda $S = \log_2 n$.

Masalan, $n = 1024$.

Ketma-ket qidiruvda $S = 512$, binar qidiruvda $S = 10$.

Agar katta xajmdagi ma'lumotlar ichida qidiruv amalga oshirilayotgan bo'lsa, u holda binar va indeksli ketma-ket qidiruvni umumlashtirib olib borish mumkin. Sababi, har ikkala qidiruv ham tartiblangan massivda amalga oshiriladi.

Nazorat savollari

1. Topilgan elementni boshiga qo'yish usulining transpozitsiya usulidan asosiy farqlari nimalardan iborat?
2. Mazkur usullar ma'lumotlar qanday ko'rinishda berilganda tezroq ishlaydi?
3. Ular qanday ro'yxatlarda ishlaydi, ya'ni tartiblangan yoki ixtiyoriy?
4. Binar qidiruvning mazmun va mohiyati nimadan iborat?
5. Qanday qilib binar daraxtni chetlab o'tish mumkin?

MA'LUMOTLAR TUZILMASI FANIDAN LABARATORIYA MASHG'ULOTLARI

1-LABARATORIYA MASHG'ULOTI: MA'LUMOTLARNING ODDIY SOZLANGAN TOIFALARI

Maqsad va vazifalar: Ma'lumotlari turlari bo'yicha bilim va ko'nikmalarni hosil qilish.

Jihozlar va dasturiy vositalar: kompyuter, Visual Studio, C++.

Ishni bajarish tartibi:

1. Ishning maqsadini tushinish.
2. Vazifaning qo'yilishi o'rganish.
3. Vazifani bajarish ketma-ketligini aniqlash.
4. Kompyuterda natijani olish.
5. Olingan natijani tahlil qilish.
6. Bajarilgan ish bo'yicha hisobot tayyorlash.

Topshiriqlar:

1. Parallelogramm yuzasini hisoblash dasturini tuzing.
2. Parallelepiped sirt yuzasini hisoblash dasturini tuzing.
3. Silindr xajmini hisoblash dasturini yozing.
4. Bir necha daftar va qalamlardan iborat xarid qiymatini hisoblash dasturini yozing.
5. Agar berilgan uchburchakning ikki tomoni va ular o'rtasidagi burchagi ma'lum bo'lsa, uchburchak yuzini hisoblash dasturini yozing.
6. Ikkita parallel ulangan karshiliklardan iborat elektr zanjir qarshiligini hisoblash dasturini yozing.
7. Avtomobilda dala hovliga borish va qaytish qiymatini hisoblash dasturini yozing.
8. Silindr sirt yuzasini hisoblash dasturini yozing.
9. Ixtiyoriy berilgan qiymatlarda parallelepiped hajmini hisoblash dasturini yozing.
10. Ixtiyoriy berilgan qiymatlarda kub xajmini hisoblash dasturini yozing.
11. Bajarilgan ishlar yuzasidan hisobot tuzing.

Tajriba ishini rasmiylashtirish va topshirish tartibi:

1. Ishning nazariy asoslari.
2. Vazifani bajarish dasturi.
3. Hisob natijalari va ularning tahlili.
4. C++ muhitida dastur matnini kiritish.
5. Seans bayonnomasi.

2-LABARATORIYA MASHG'ULOTI: MANTIQUIY MA'LUMOTLAR TURLARI. MASSIV ELEMENTLARI

Maqsad va vazifalar: Ma'lumotlari turlari bo'yicha bilim va ko'nikmalarni hosil qilish.

Jihozlar va dasturiy vositalar: kompyuter, Visual Studio, C++.

Ishni bajarish tartibi:

1. Ishning maqsadini tushinish.
2. Vazifaning qo'yilishi o'rganish.
3. Vazifani bajarish ketma-ketligini aniqlash.
4. Kompyuterda natijani olish.
5. Olingan natijani tahlil qilish.
6. Bajarilgan ish bo'yicha hisobot tayyorlash.

Topshiriqlar:

1. Matritsani matritsaga ko'paytiring
2. Jadval hosil qiling va unga ma'lumotlarni kiriting, ekranga chiqaring.
3. Talabalar ism-familiyasi, yoshi va ballaridan iborat jadval yarating va talabalarni ism-familiyasini alfavit bo'yicha tartibga keltiring.
4. 2 ta bir xil tipdagi jadval berilgan. Ikkala jadvalni o'zaro solishtiring va aynan bir xil bo'lgan yozuvlarni o'chiring.
5. Birlashma tipdagi (tipda 2 ta element e'lon qiling) 2 ta o'zgaruvchini bir-biridan farqli elementlariga qiymat bering va ularning yig'indisini ikkala o'zgaruvchining qiymat berilmagan elementlariga o'zlashtiring. Har ikkala o'zgaruvchining barcha elementlarini ekranga chiqaring va natijani tushuntiring.
6. Berilgan matnli fayldan simvollarni o'qib, ekranga chiqaring va raqamlarni ajratib ko'rsating.
7. Berilgan matnli faylda a harfi necha marta qatnashganini sanang.
8. Berilgan matnli fayldagi satrlar sonini aniqlang.
9. Talabalar ism-familiyasi, yoshi va ballari maydonidan iborat klass yarating va talabalar ro'yhatini tuzing. So'ralayotgan talaba ro'yhatda bor yo'qligini aniqlang.

- 10.Oy nomlaridan iborat sanaladigan toifa yarating. So'ralayotgan oy qaysi faslga tegishligini aniqlang.
- 11.Mahsulot nomlaridan iborat elementlar va ularning qiymati sifatida narxlari kiritilgan sanaladigan toifa yarating. So'ralayotgan narxda qanday mahsulot yoki mahsulotlar to'plamini xarid qilsa bo'ladi, shuni aniqlash dasturini tuzing.
- 12.Kiritilgan ismning harflarini alfavit bo'yicha tartibga keltiring.
- 13.Satrlı toifadagi vektor berilgan. Bir xil qiymatdagi elementlarni aniqlang, ekranga chiqaring.

Tajriba ishini rasmiylashtirish va topshirish tartibi:

1. Ishning nazariy asoslari.
2. Vazifani bajarish dasturi.
3. Hisob natijalari va ularning tahlili.
4. C++ muhitida dastur matnini kiritish.
5. Seans bayonnomasi.

3-LABARATORIYA MASHG'ULOTI:YOZUVLAR USTIDA BAJARILADIGAN AMALLAR.

Maqsad va vazifalar: Standart murakkab turlar bilan ishlashda vektor, yozuvlar tugrisida

amaliy

bilim va ko'nikmalar hosil qilish.

Jihozlar va dasturiy vositalar: Kompyuter, Windows operatsion tizimi, C++ dasturi, videoproektor.

Topshiriqlar:

1. Foydalanuvchidan oy rakamini so'rovchi va bu oy yilning kaysi mavsumiga tegishli ekanligi to'grisidagi javob beruvchi dastur tuzing. Agar foydalanuvchi mumkin bo'lmagan son kiritrsa, dastur xato son

berilganligi to'grisida axborot chikarsin. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan. Oy raqamini kiriting (1 dan 12 gacha) va <Enter> ni bosing ® 11

2. Shaxarlararo telefon orkali so'zlashuv qiymatini xisoblovchi dastur tuzing (u minut baxosi - abonent turining shaxargacha bo'lgan masofasi - uzunligi bilan aniklanadi). Dastur uchun shaxar kodi va so'zlashuv vaqti boshlang'ich qiymatlar bo'lib xizmat qiladi. Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Telefon orqali so'zlashuvni hisoblash.

Boshlang'ich qiymatlarni kiriting:

Shahar kodi ® 812

So'zlashuv vaqti (butun son minut) ® 3

Shahar: Toshkent

Minut bahosi: 150 so'm

So'zlashuv vaqti: 450 so'm

Shahar	Kod	Bir minut bahosi, so'm
Toshkent	371	150
Samarqand	322	120
Qarshi	375	100
Buxoro	365	130

3. Keyingi kun sanasini hisoblovchi dastur tuzing.

Quyida dastur ishlash vaqtida tavsiya qilinadigan ekran ko'rinishi berilgan.

Bugungi sanani raqamlarda kiriting (son oy yil) ® **31 12 2002**.

Oyning oxirgi kuni! Kelayotgan yangi yil bilan! Ertaga 1.1.2003.

4. Foydalanuvchidan hafta kuni raqamini so'rovchi va shu kunning nomlanishini ekranga chiqaruvchi dastur tuzing.

5. Birinchi o'nta butun musbat sonlar kvadratlari jadvalini chiqaruvchi dastur tuzing. Quyida dastur ishlash vaqtida hosil bo'lishi kerak bo'ladigan ekran ko'rinishi tavsiya etiladi:

Kvadratlar jadvali

Son	Kvadrat
1	1
2	4
3	9
4	16
5	25

Son	Kvadrat
6	36
7	49
8	64
9	81
10	100

6. Birinchi «n» ta butun musbat sonlar yig'indisini hisoblovchi dasturni yozing.
7. Birinchi butun musbat toq sonlar kvadratlari jadvalini chiqaruvchi dastur tuzing.
8. Dastlabki n ta butun musbat sonlar yig'indisini hisoblovchi dastur tuzing.
9. 1,3,5,7, ... qatorning birinchi n ta hadi yig'indisini hisoblash dasturini tuzing.
10. $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots$ qatorning birinchi n ta hadi yig'indisini hisoblash dasturini tuzing.

Tajriba ishini rasmiylashtirish va topshirish tartibi:

1. Ishning maqsadini tushinish.
2. Vazifaning qo'yilishi o'rganish.
3. Vazifani bajarish ketma-ketligini aniqlash.
4. Kompyuterda natijani olish.

5. Olingan natijani tahlil qilish.
6. Bajarilgan ish bo'yicha hisobot tayyorlash.

Hisobotning mazmuni

1. Ishning nazariy asoslari.
2. Vazifani bajarish dasturi.
3. Hisob natijalari va ularning tahlili.
4. C++ muhitida dastur matnini kiritish.
5. Seans bayonnomasi.

4-LABARATORIYA MASHG'ULOTI:MA'LUMOTLARNING STEK TURLARI.

Maqsad va vazifalar: Ro'yxatlar, stek va navbat haqida umumiy tushuncha hosil qilish.

Jihozlar va dasturiy vositalar: Kompyuter, Windows operatsion tizimi, C++ dasturi, videoproektor.

Topshiriqlar:

1. Stekda element borligini tekshirish
2. Stekga elementni joylashtirish
3. Stekdan elementni ajratib olish
4. To'lalikka tekshirish
5. Yuqori elementni aniqlash
6. Navbatga element qo'yish
7. Navbatdan elementni chiqarib tashlash
8. Navbatni bo'shligini tekshirish
9. Dekdan elementni chiqarib tashlash
10. Dekka element qo'yish

Tajriba ishini rasmiylashtirish va topshirish tartibi:

1. Ishning maqsadini tushinish.
2. Masalani quyilishi o'rganish.
3. Masalani echish ketma-ketligini aniqlash.
4. C++ muhitida dastur matnini kiritish.
5. Kompyuterda natijani olish.
6. Olingan natijani tahlil qilish.
7. Bajirilgan ish bo'yicha hisobot tayyorlash.

Hisobotning mazmuni

1. Ishning nazariy asoslari.
2. Masalani echish dasturi.
3. Hisob natijalari va olingan natijalarni taxlili.
4. C++ muhitida dastur matnini kiritish.
5. Seans bayonnomasi.

5-LABARATORIYA MASHG'ULOTI:MA'LUMOTLARNING STEK TURLARI

Maqsad va vazifalar: Ro'yxatlar, stek va navbat haqida umumiy tushuncha hosil qilish.

Jihozlar va dasturiy vositalar: Kompyuter, Windows operatsion tizimi, C++ dasturi, videoproektor.

Topshiriqlar:

1. Stek birinchi va oxirgi elementlari o'ni almashtirilsin.
2. Stek elementlari teskari tartibda joylashtirib chiqilsin.
3. Stek o'rtasidagi element o'chirib tashlansin.
4. Stekning juft o'rinda turgan elementlari

o'chirilsin.

5. Stek o'rtasiga '*' belgi joylashtirilsin.

6. Stek eng kichik elementi topilsin va undan keyin 0 joylashtirilsin.

7. Stek eng katta elementi topilsin va undan keyin 0 joylashtirilsin.

8. Stek eng kichik elementi o'chirilsin.

9. Stekda birinchi elementga teng barcha elementlar o'chirilsin.

10. Stek oxirgi elementiga teng barcha elementlar o'chirilsin.

Tajriba ishini rasmiylashtirish va topshirish tartibi:

1. Ishning maqsadini tushinish.
2. Masalani quyilishi o'rganish.
3. Masalani echish ketma-ketligini aniqlash.
4. C++ muhitida dastur matnini kiritish.
5. Kompyuterda natijani olish.
6. Olingan natijani tahlil qilish.

Hisobotning mazmuni

1. Ishning nazariy asoslari.
2. Masalani echish dasturi.
3. Hisob natijalari va olingan natijalarni taxlili.
4. C++ muhitida dastur matnini kiritish.
5. Seans bayonnomasi.

6-LABARATORIYA MASHG'ULOTI:MA'LUMOTLARNING NAVBAT TURLARI.

Maqsad va vazifalar: Ro'yxatlar, stek va navbat haqida umumiy tushuncha hosil qilish.

Jihozlar va dasturiy vositalar: Kompyuter, Windows operatsion tizimi, C++ dasturi, videoproektor.

Topshiriqlar:

- 1.Navbatda birinchi va oxirgi elementlar o'rnini almashtirilsin.
2. Navbat o'rtasidagi element o'chirib tashlansin. Agar navbat elementlari soni toq bo'lsa, bitta element, aks holda ikkita element o'chirilsin.
3. Navbatni juft o'rinda turgan elementlari o'chirilsin.
4. Navbat o'rtasiga '+' belgi joylashtirilsin.
5. Navbat eng kichik elementi topilsin va undan keyin 0 joylashtirilsin. 6. Navbat eng katta elementi topilsin va undan keyin 0 joylashtirilsin.
7. Navbat eng kichik elementi o'chirilsin.
8. Navbatda birinchi elementga teng barcha elementlar o'chirilsin.
9. Navbatda oxirgi elementga teng barcha elementlar o'chirilsin.
10. Navbat eng katta elementi o'chirilsin.
11. Navbat eng kichik elementi topilsin va uning o'rniga 0 joylashtirilsin.

Tajriba ishini rasmiylashtirish va topshirish tartibi:

1. Ishning maqsadini tushinish.
2. Masalani quyilishi o'rganish.
3. Masalani echish ketma-ketligini aniqlash.
4. C++ muhitida dastur matnini kiritish.
5. Kompyuterda natijani olish.
6. Olingan natijani tahlil qilish.
7. Bajarilgan ish bo'yicha hisobot tayyorlash.

Hisobotning mazmuni

1. Ishning nazariy asoslari.
2. Masalani echish dasturi.
3. Hisob natijalari va olingan natijalarni taxlili.
4. C++ muhitida dastur matnini kiritish.
5. Seans bayonnomasi.

7-LABARATORIYA MASHG'ULOTI: DINAMIK MA'LUMOTLAR TUZILMASI. BIR BOG'LAMLI RO'YHAT BOSHIDAN ELEMENTNI O'CHIRISH.

Maqsad va vazifalar: Ro'yxatlar bilan ishlash bilim va malakalarni hosil qilish.

Jihozlar va dasturiy vositalar: Kompyuter, C++ dasturi

Ishni bajarish tartibi:

1. Ishning maqsadini tushinish.
2. Masalani qo'yilishi o'rganish.
3. Masalani echish ketma-ketligini aniqlash.
4. Masalani dasturini tuzish.
5. Pascal muhitida dastur matnini kiritish.
6. Kompyuterda natijani olish.

7. Olingan natijani tahlil qilish.
8. Bajarilgan ish bo'yicha hisobot tayyorlash.

Topshiriqlar

1. Ro'yxat elementini yaratish
2. Ro'yxatda elementni qidirish
3. Ro'yxatning ko'rsatilgan joyiga elementni qo'yish
4. Berilgan elementni ro'yxatdan o'chirish.
5. Navbatdan elementni o'chirish
6. GetNode amali
7. FreeNode amali

Tajriba ishini rasmiylashtirish va topshirish tartibi:

1. Ishning nazariy asoslari.
2. Masalani echish dasturi.
3. Hisob natijalari va olingan natijalarni taxlili.
4. C++ muhitida dastur matnini kiritish.
5. Seans bayonnomasi.

8-LABARATORIYA MASHG'ULOTI: BINAR DARAXTLARNI QURISH. BINAR DARAXT YARATISH FUNKSIYASI.

Ishning maqsadi: Daraxtlarlar bilan ishlash

Jihozlar va dasturiy ta'minotlar: Kompyuter, operatsion tizim, C++ dasturlash tili.

Topshiriqlar

1. Berilgan binar daraxtning oraliq tugunlaridan tashkil topgan yangi binar daraxt tuzish algoritmi va dasturini keltiring.
2. Berilgan binar daraxtdan kalitlari o'sish tartibida joylashgan bir bog'lamli ro'yxat hosil qilish algoritmi va dasturini keltiring.

3. Binar daraxtning barcha barglari yozuvini chop etuvchi dastur ishlab chiqing.

4. Binar daraxtning barcha oraliq tugunlari yozuvini chop etuvchi dastur ishlab chiqing.

5. Binar daraxtning juft qiymatli kalitga ega elementlaridan yangi daraxt qurish algoritmi va dasturini keltiring.

6. Binar daraxtning tugunlari sonini aniqlashning algoritmi va dasturini keltiring.

Tajriba ishini bajarilish tartibi.

1. Ishning maqsadini tushinish.
2. Masalani qo'yilishi o'rganish.
3. Masalani echish ketma-ketligini aniqlash.
4. Masalani dasturini tuzish.
5. C++ muhitida dastur matnini kiritish.
6. Kompyuterda natijani olish.
7. Olingan natijani tahlil qilish.
8. Bajarilgan ish bo'yicha hisobot tayyorlash.

Hisobotning mazmuni

1. Ishning nazariy asoslari.
2. Masalani echish dasturi.
3. Hisob natijalari va olingan natijalarni taxlili.
4. C++ muhitida dastur matnini kiritish.
5. Seans bayonnomasi.

9-LABARATORIYA MASHG'ULOTI: MA'LUMOTLARNI DARAXTSIMON TAQDIM ETISHDA

Ishning maqsadi: Daraxtlarlar bilan ishlash

Jihozlar va dasturiy ta'minotlar: Kompyuter, operatsion tizim, C++

dasturlash tili.

Topshiriqlar

1. Talabalar ismlari ketma-ketligidan binar daraxt hosil qilish algoritmi va dasturini tuzing.
2. Berilgan binar daraxtning terminal tugunlaridan tashkil topgan yangi muvozanatlangan binar daraxt hosil qilish algoritmi va dasturini tuzing.
3. Berilgan binar daraxtning har bir tuguni chap tomoni tugunlaridan tashkil topgan muvozanatlangan binar daraxt hosil qilish algoritmi va dasturini tuzing.
4. Daraxt tugunlari haqiqiy sonlar bo'lsin. Daraxt barcha tugunlarini o'rta arifmetigini hisoblash algoritmi va dasturini keltiring.
5. Daraxt tugunlari haqiqiy sonlar bo'lsin. Yozuvi manfiy bo'lgan daraxt tugunlarini o'chiruvchi dastur tuzing.
6. Daraxt tugunlari haqiqiy sonlar bo'lsin. Yozuvi berilgan kalit qiymatidan katta bo'lgan daraxt tugunlarini o'chiruvchi dastur tuzing.
7. Berilgan binar daraxtning balandligini aniqlash algoritmi va dasturini keltiring.

Tajriba ishini bajarilish tartibi.

1. Ishning maqsadini tushinish.
2. Masalani qo'yilishi o'rganish.
3. Masalani echish ketma-ketligini aniqlash.
4. Masalani dasturini tuzish.
5. C++ muhitida dastur matnini kiritish.
6. Kompyuterda natijani olish.
7. Olingan natijani tahlil qilish.
8. Bajarilgan ish bo'yicha hisobot tayyorlash.

Hisobotning mazmuni

1. Ishning nazariy asoslari.
2. Masalani echish dasturi.
3. Hisob natijalari va olingan natijalarni taxlili.
4. C++ muhitida dastur matnini kiritish.
5. Seans bayonnomasi.

10-LABARATORIYA MASHG'ULOTI:TASHQI SARALASH

Maqsad va vazifa: Tashqi saralash algoritmlari bilan tanishish.

Jihozlar va dasturiy vositalar: kompyuter, MS Visual Studio,C ++ dasturi.

Topshiriqlar:

1. Input.txt faylida berilgan 1-katordagi suzlarni usish tartibida saralab Output.txt fayliga natijani yozing.
2. Input.txt faylida berilgan 1-katordagi suzlarni kamayish tartibida saralab Output.txt fayliga natijani yozing.
3. Input.txt faylida berilgan matritsani kamayish tartibiga nisbatan saralab Output.txt fayliga natijani yozing.
4. Input.txt faylida berilgan matritsani usish tartibiga nisbatan saralab Output.txt fayliga natijani yozing.
5. nxm matritsaning 3 ga bo'linadigan elementlarini kamayish tartibiga nisbatan saralang.

Ishni bajarish tartibi:

1. Ishning maqsadini tushinish.
2. Vazifaning qo'yilishi o'rganish.
3. Vazifani bajarish ketma-ketligini aniqlash.
4. C++ muhitida dastur matnini kiritish.
5. Kompyuterda natijani olish.
6. Olingan natijani tahlil qilish.

7. Bajarilgan ish bo'yicha hisobot tayyorlash.

Hisobotning mazmuni

1. Ishning nazariy asoslari.
2. Vazifani bajarish dasturi.
3. Hisob natijalari va ularning tahlili.
4. Seans bayonnomasi.

11-LABARATORIYA MASHG'ULOTI:SARALASH USULLARINI TANLASHDA HISOBGA OLINADIGAN OMILLAR. YOZUVNI JOYLASHTIRISH USULLARI

Maqsad va vazifa: Ichki saralash algoritmlari bilan tanishish.

Jihozlar va dasturiy vositalar: kompyuter, MS Visual Studio,C ++ dasturi.

Topshiriqlar:

1. Mashina egalarining ismlari bo'yicha alifbo tartibida joylashtirilsin va mos ravishda ularning mashinalari haqidagi ma'lumotlar chiqarilsin.

2. Avtomobillarni ta'mirlash tartibi ishlab chiqilsin. Bu yerda ta'mir tugashi sanasi qaysi avtomobil uchun ertaroq bo'lsa, shunga birinchi navbatda xizmat ko'rsatiladi.

3. Oldingi ta'mir qilinganlar soni 2 ga teng bo'lgan mashinalar raqamlari bo'yicha kamayish tartibida joylashtirilsin.

4. Oldin ta'mir qilinmagan mashinalarni ta'mirdan chiqish sanasi bo'yicha o'sish tartibida joylashtiring.

5. 'Mercedes' markali mashina egalarini alifbo bo'yicha teskari tartibda joylashtiring.

6. Boshqalaridan oldinroq ta'mirlanadigan mashinalarni ularning markasi bo'yicha alifbo tartibida joylashtiring (ta'mir tugatilishi sanasi 31.12.2012 dan erta).

7. 'Nexia' markasidagi mashinalarni raqamlari bo'yicha o'sish tartibida joylashtiring.

8. O'tgan yildan beri ta'mirlanmagan mashinalarni ularning egalari ismlari bo'yicha alifbo tartibida joylashtiring.

9. Keyingi oyda ta'mirlanishi lozim bo'lgan mashinalarni oxirgi marta ta'mirlanganlik sanasi bo'yicha o'sish tartibida keltiring.

10. 'Mercedes' markasidagi
tartibida joylashtiring.

Ishni bajarish tartibi:

8. Ishning maqsadini tushinish.
9. Vazifaning qo'yilishi o'rganish.
10. Vazifani bajarish ketma-ketligini aniqlash.
11. C++ muhitida dastur matnini kiritish.
12. Kompyuterda natijani olish.
13. Olingan natijani tahlil qilish.
14. Bajarilgan ish bo'yicha hisobot tayyorlash.

Hisobotning mazmuni

5. Ishning nazariy asoslari.
6. Vazifani bajarish dasturi.
7. Hisob natijalari va ularning tahlili.
8. Seans bayonnomasi.

12-LABARATORIYA MASHG'ULOTI: KETMA-KET IZLASH

Mashg'ulot maqsada: ketma –ket izlash algoritmlari bilan tanishish

Mashg'ulotda qo'llaniladigan jihozlar: Doska, EHM, C++ dasturlash tili.

Topshiriqlar

1. Ketma-ket qidiruv usulidan foydalanib, ro'yhat eng kichik elementini toping.

2. Ketma-ket qidiruv usulidan foydalanib, ro'yhatda berilgan kalitdan katta elementlarni toping.

3. Ketma-ket qidiruv usulidan foydalanib, ro'yhat eng kichik elementini toping.

4. Ketma-ket va binar qidiruv usulidan foydalanib, A massivdan elementni va taqqoslashlar sonini toping.

5. Binar qidiruvdan foydalanib elementlarni tasodifiy ravishda toping.

6. Mashina raqamlari ro'yhati berilgan: 345, 368, 876, 945, 564, 387, 230. Binar qidiruvdan foydalanib berilgan raqamli mashina qaysi joyda turganini toping.

7. Ketma-ket qidiruv usulidan foydalanib ro'yhatda har ikkinchi elementni qidiring va taqqoslashlar sonini aniqlang.

Tajriba ishini bajarilish tartibi

1. Ishning maqsadini tushinish.
2. Vazifaning qo'yilishi o'rganish.
3. Vazifani bajarish ketma-ketligini aniqlash.
4. C++ muhitida dastur matnini kiritish.
5. Kompyuterda natijani olish.
6. Olingan natijani tahlil qilish.
7. Bajarilgan ish bo'yicha hisobot tayyorlash.

Hisobotning mazmuni

1. Ishning nazariy asoslari.
2. Vazifani bajarish dasturi.
3. Hisob natijalari va ularning tahlili.
4. Seans bayonnomasi.

13-LABARATORIYA MASHG'ULOT:IKKILANGAN DARAXT BO'YICHA IZLASH.

Mashg'ulotda qo'llaniladigan jihozlar: Doska, EHM, C++ dasturlash tili.

Topshiriqlar

1. Berilgan ro'yhatda qidirilayotgan element transpozitsiya usuli bilan qancha murojaatda ro'yhat boshiga kelishini aniqlash dasturini tuzing.
2. Massivdan boshiga qo'yish usuli yordamida key kalitli elementni izlash dasturini tuzing.
3. Binar qidiruv usuli yordamida massivga yangi elementni kiriting.
4. Binar qidiruv usuli yordamida massivning key kalitli elementini o'chiring.
5. Ro'yhatda transpozitsiya usuli yordamida toq elementlarni topish dasturini tuzing.
6. Berilgan massivda key kalitli elementni ketma-ket va binar qidiruv usullari yordamida izlang va qaysi usul ushbu qidiruv holatida samara berganligini aniqlash dasturini keltiring.
7. Talabalar ismi va umumiy ballaridan iborat jadvaldan ketma-ket qidiruv usuli bilan balli maksimal bo'lgan talabani toping.
8. Talabalar ismi va umumiy ballaridan iborat jadvaldan binar qidiruv usuli yordamida so'ralgan talabaning umumiy balini chiqarish dasturini tuzing.
9. Boshiga qo'yish usuli yordamida talabalar ismlaridan iborat massiv elementlariga ko'p marta murojaat qilib massivni qayta tartiblang.
10. Transpozisiya usuli yordamida talabalar ismlaridan iborat ro'yhat elementlariga ko'p marta murojaat qilib massivni qayta tartiblang.

Tajriba ishini bajarilish tartibi

1. Ishning maqsadini tushinish.
2. Vazifaning qo'yilishi o'rganish.
3. Vazifani bajarish ketma-ketligini aniqlash.

4. C++ muhitida dastur matnini kiritish.
5. Kompyuterda natijani olish.
6. Olingan natijani tahlil qilish.
7. Bajarilgan ish bo'yicha hisobot tayyorlash.

Hisobotning mazmuni

1. Ishning nazariy asoslari.
2. Vazifani bajarish dasturi.
3. Hisob natijalari va ularning tahlili.
4. Seans bayonnomasi.

14-LABARATORIYA MASHG'ULOTI:MA'LUMOTLARDAN BEVOSITA ERKIN FOYDALANADIGAN IZLASH USULI

Mashg'ulot maqsada: erkin foydalaniladigan izlash algoritmlari bilan tanishish

Mashg'ulotda qo'llaniladigan jihozlar: Doska, EHM, C++ dasturlash tili.

Topshiriqlar

1. Binar qidiruvdan foydalanib elementlarni tasodifiy ravishda toping.
2. Mashina raqamlari ro'yhati berilgan: 345, 368, 876, 945, 564, 387, 230.
Binar qidiruvdan foydalanib berilgan raqamli mashina qaysi joyda turganini toping.
3. Ketma-ket qidiruv usulidan foydalanib ro'yhatda har ikkinchi elementni qidiring va taqqoslashlar sonini aniqlang.
4. Binar qidiruvdan foydalanib massivdan berilgan kalitga karrali kalitli elementni va solishtirishlar sonini toping.

Tajriba ishini bajarilish tartibi

1. Ishning maqsadini tushinish.
2. Vazifaning qo'yilishi o'rganish.
3. Vazifani bajarish ketma-ketligini aniqlash.
4. C++ muhitida dastur matnini kiritish.

5. Kompyuterda natijani olish.
6. Olingan natijani tahlil qilish.
7. Bajarilgan ish bo'yicha hisobot tayyorlash.

Hisobotning mazmuni

1. Ishning nazariy asoslari.
2. Vazifani bajarish dasturi.
3. Hisob natijalari va ularning tahlili.
4. Seans bayonnomasi.

MA'LUMOTLAR TUZILMASI FANIDAN AMALIY MASHG'ULOTLAR

GLOSSARIY

Amaliy dasturlar paketi (ADP) – bu muayyan (funktsional tizimosti, biznes - ilova) sinf vazifalarini hal etish uchun milljallangan dasturlar majmui.

Axborot jamiyati – ko'pchilik ishlovchilarning axborot, ayniqsa uning oliy shakli bo'lmish bilimlarni ishlab chiqarish, saqlash, qayta ishlash va amalga oshirish bilan band bo'lgan jamiyatidir.

Axborot texnologiyasi - axborotni to'plash, saqlash, izlash, unga ishlov berish va uni tarqatish uchun foydalaniladigan uslublar, qurilmalar, usullar va jarayonlar majmui.

Axborot zaxiralari– aloxida xujjat va aloxida xujjat to'plami, axborot tizimlari (kutubxona, arxiv, fond, ma'lumotlar banklari, boshqa axborot tizimlari) dagi xujjatlar va xujjatlar to'plamidir.

Axborot infratuzilmasi- axborot tizimlarining ishlashini ta'minlovchi tashkiliy-texnikaviy tuzilmalar va ob'ektlar, ma'lumotlar banklari va bazalari, aloqa va ma'lumotlar uzatish tizimlarining majmui.

Axborotlashtirish-axborot resurslari, axborot texnologiyalari va axborot

tizimlaridan foydalanilgan holda yuridik va jismoniy shaxslarning axborotga boʻlgan ehtiyojlarini qondirish uchun shart-sharoitlar yaratishning tashkiliy ijtimoiy-iqtisodiy va ilmiy-texnikaviy jarayoni.

Axborot resursi- moddiy jismlardagi yoki axborot tizimi tarkibidagi axborot, ma'lumotlar banki va bazasi, dasturiy mahsulot.

Axborot resurslarining, axborot tizimlarining mulkdori – axborot resurslariga, axborot tizimlariga toʻla hajmda egalik qiluvchi, ulardan foydalanuvchi va ularni tasarruf etuvchi yuridik yoki jismoniy shaxs.

Axborot resurslarining, axborot tizimlarining egasi – axborot resurslariga, axborot tizimlariga qonun bilan yoki axborot resurslarining, axborot tizimlarining mulkdori tomonidan belgilangan xuquqlar doirasida egalik qiluvchi, ulardan foydalanuvchi va ularni tasarruf etuvchi yuridik yoki jismoniy shaxs.

Axborot tizimi – axborot resurslari, axborot texnologiyalari va aloqa vositalarining axborotni toʻplash, saqlash, izlash, unga ishlov berish va undan foydalanish imkonini beradigan tashkiliy jihatdan tartibga solingan majmui.

Axborot mulkdori – oʻz mablagʻiga yoki boshqa qonuniy yoʻl bilan olingan axborotga egalik qiluvchi, undan foydalanuvchi va uni tasarruf etuvchi yuridik yoki jismoniy shaxs.

Axborotni muxofaza etish – axborot borasidagi havfsizlikka taxdidlarning oldini olish va ularning oqibatlarini bartaraf etish chora-tadbirlari.

Axborot soxasi – sub'ektlarning axborotni yaratish, qayta ishlash va undan foydalanish bilan bogʻliq faoliyati soxasi.

Axborot borasidagi havfsizlik – axborot soxasida shaxs, jamiyat va davlat manfaatlarining himoyalanganlik holati.

Axborot egasi – qonunda yoki axborot mulkdori tomonidan belgilangan xuquqlari doirasida axborotga egalik qiluvchi, undan foydalanuvchi va uni tasarruf etuvchi yuridik yoki jismoniy shaxs.

Axborot xavfsizligini ta'minlash – bu axborotlashgan munosabatlar sub'ektlari huquq va qiziqishlarini qanoatlantirish kafolatidir.

Videomonitor (display) - ShK ga kiritiladigan va undan chiqadigan axborotni aks ettiruvchi moslamadir

Dasturiy interfeys – hisoblash tizimi doirasida qurilma va dasturlar o'zaro ta'sirini ta'minlovchi vositalar yig'indisi.

Duragay hisoblash mashinalari (DHM) – kombinatsiyalashgan holda amal qiluvchi hisoblash mashinalari bo'lib, hamda raqamli ham o'xshashli shaklda taqdim etilgan axborot bilan ishlaydi; ular UXM va RXMning afzalliklarini o'zida jo etgan.

Elektron raqamli imzo – elektron hujjatning mazkur elektron hujjat axborotini elektron raqamli imzoning yopiq kalitidan foydalangan holdagi maxsus o'zgartirishlar natijasida olingan hamda elektron raqamli imzoning ochiq kaliti yordamida elektron hujjatdagi axborotda buzilishlar yo'qligini aniqlash va elektron raqamli imzo yopiq kalitining egasini identifikatsiyalash imkoniyatini beradigan imzosi.

Elektron raqamli imzoning yopiq kaliti – elektron raqamli imzo vositalaridan foydalangan holda olingan, faqat imzo qo'yuvchi shaxsga ma'lum bo'lgan va elektron hujjatda elektron raqamli imzo tuzish uchun mo'ljallangan ramzlar ketma-ketligi.

Elektron raqamli imzoning ochiq kaliti – elektron raqamli imzo vositalaridan foydalangan holda olingan, elektron raqamli imzoning yopiq kalitiga mos keluvchi, axborot tizimining har qanday foydalanuvchisi foydalana oladigan va elektron hujjatdagi elektron raqamli imzoning haqiqiylikini tasdiqlash uchun mo'ljallangan ramzlar ketma-ketligi.

Elektron raqamli imzoning haqiqiylikini tasdiqlash – elektron raqamli imzoninig elektron raqamli imzo yopiq kalitining egasiga tegishlilikini va elektron hujjatda buzilishlar yo'qligini tekshirishning ijobiy natijasi.

Informatika – kompyuterlar yordami va ularni qo'llash muxiti vositasida axborotni yangilash jarayonlari bilan bog'liq inson faoliyati sohasidir.

Ishchi stansiyalar – muayyan turdagi (grafik, muxandislik, nashriyot va boshqalar) ishlarni bajarishga ixtisoslashtirilgan bir kishi foydalanadigan qudratli mikro - EHM lardir.

Kibernetika – texnik , biologik, ijtimoiy va boshqa turli tizimlarda boshqaruvning umumiy tamoyillari xaqidagi fandır.

Klaviatura – son bilan ko'rsatiladigan, matnli va boshqariluvchi axborotni ShKga qo'l yordamida kirishi uchun mo'ljallangan moslama.

Qobiq – biror bir dastur va foydalanuvchi o'rtasidagi qatlam yoki boshqa dastur ustida usqurtma bo'lgan dastur.

Masofaviy ta'lim – o'qitishning universal shakli sifatida, yangi axborot va telekommunikatsiya texnologiyalariga, texnik vositalarning keng spektrlaridan foydalanishga asoslangan bo'lib, o'quvchilar tomonidan o'qitish darslarini erkin tanlash, o'qituvchi bilan muloqot qilish imkoniyatlarini ta'minlaydi.

Ma'lumotlar bazasi deganda - axborot ishlari uchun, ya'ni o'zaro bog'langan katta hajmdagi axborotlarni saqlash, o'zgartirish, qayta ishlash, hisobotlarni olish uchun maxsus tizim tushuniladi va u mashinadan tashqari axborot ta'minotining tizimlari yordamida hosil qilinadi.

Ma'lumotlar bazasini boshqarish tizimi - dasturiy va til vositalarining t'plamidan iborat bo'lib, ular yordamida MB ni hosil qilish, yuritish, tahrirlash va boshqa vazifalarni bajarish mumkin.

Maxfiy axborot – foydalanishi qonun hujjatlariga muvofiq cheklab qo'yiladigan hujjatlantirilgan axborot.

Mijoz (klient) – vazifa, ishchi stantsiya yoki kompyuter tarmog'idan foydalanuvchi.

Multimedia vositalari (multimedia - ko'pvositalilik)- informatikaning

dasturiy va texnikaviy vositalari asosida audio, video, matn, grafika va animatsiya (ob'ektlarning fazodagi harakati) effektlari asosida o`quv materiallarini o`quvchilarga etkazib berishning mujassamlangan holdagi ko`rinishidir.

Operatsion tizim (OT) – bu EHM zaxiralarini boshqarish, amaliy dasturlarni chiqarish va ularning tashqi qurilmalar, boshqa dasturlar bilan o'zaro aloqasini amalga oshiruvchi, shuningdek, foydalanuvchining kompyuter bilan muloqotini ta'minlovchi dasturiy vositalar yig'indisidir.

Ommaviy axborot – cheklanmagan doiradagi shaxslarga mo'ljallangan hujjatlashtirilgan axborot, bosma, audiovizual hamda boshqa xabarlar va materiallar.

Raqamli hisoblash mashinalari (RHM) – diskretli ishlaydigan hisoblash mashinalari bo'lib, diskret, aniqrog'i raqamli shaklda taqdim etilgan axborot bilan ishlaydi.

Serverlar – barcha stantsiya tarmoqlaridan olingan surovlarni qayta ishlash uchun ajratilgan ko'p kishi foydalanadigan qudratli mikro EHMLardir.

Dispatcher- bu chiqarish mexanizmining bir qismi, u predmetli bilimlardan qoidalarni qachon va qanday tartibda qo'llashini hal qiladi.

Intellektual ADP - ilmiy va iqtisodiy faoliyat bilan bog'liq sohalarga keng tarqalgan dasturiy vositalar.

Interpreter - ekspert tizimining bir qismi bo'lib, bazadagi bilimlarni ma'lum bir tartibda qayta ishlaydi.

Interfeys - bu bilimlar bazasi tizimining bir qismi bo'lib, u bilimlar bazasi va qarorlarni olish mexanizmi bilan ishni bilimlar bazasiga tegishli bo'lgan predmetli soha mutaxassislarining kasbiy tiliga yaqin bo'lgan etarlicha yuqori darajadagi tilni ta'minlaydi.

Ishonishlik koeffitsienti - bu ehtimollik yoki ishonchlilik darajasini

bildiruvchi son, u bilan ushbu dalillar va qoidalarni ishonchli deb hisoblash mumkin.

Kvozidnamik ET- vaqtning ba'zi bir qayd etilgan oraligida o'zgaradigan vaziyatlarni interpritatsiyalaydi.

Qoida - bu bilimlar vazifasining AGAR <Shart>, UNDA <harakat> ko'rinishidagi rasmiy usuli.

Loyihalashtirish - oldindan belgilangan xususiyatlarga ega "ob'ektlar"ni hozirlashga ixtisoslikni tayyorlashdan iboratdir.

Mantiqiy modellar - bilimlarning alohida birliklari (dalillar) o'rtasidagi munosabatlar rasmiy nazariyaning sintaktik bilimlari yordamida aks ettiradi.

Mantiqiy xulosalar bloki - qoidalarni faktlar bilan qiyoslagan holda xulosalar mantiqini yuzaga keltirish.

Ma'lumotlar - joriy paytda echilayotgan vazifa haqidagi dastlabki, oraliq va yakuniy (natijaviy) axborotlar.

Monitoring- vaqtning haqiqiy qo'llashida ma'lumotlarni uzluksiz interpritatsiyalangan va u yoki bu o'lchamlarni Yo'l qo'yiladigan chegaradan tashqariga chiqishi haqida xabar berishdir.

O'rgatuvchi ekspert tizimlari - foydalanuvchilarga berilgan sohada tashhis qo'yish va tahlil etish imkoniyatini beradigan tizim.

Predmetli sohaning qayta o'zgartirish vazifasi - bu echimda predmetli sohaning

Bitta holati boshqasiga qayta o'zgartiriluvchi vazifalardir.

KIRISH

Ushbu dastur O'zbekiston respublikasining "Ta'lim to'g'risida"gi qonuni hamda "Kadrlar tayyorlash Milliy dasturi"ga muvofiq ishlab chiqildi.

Mazkur ishchi o'quv fan dasturi bakalvriat **yo'nalishi 5111000 – Kasb ta'limi (5330200 – Informatika va axborot texnologiyalari)** ta'lim yo'nalishida o'qiladigan "Ma'lumotlar tuzilmasi" o'quv fani bo'yicha tuzilgan bo'lib, bo'lajak fan o'qituvchisi egallashi kerak bo'lgan qo'yidagi bilimlar va ko'nikmalar majmuini o'z ichiga oladi.

Fanning maqsad va vazifalari

Fanning o'qilishidan maqsad – informatika sohasida qo'llaniladigan ma'lumotlarning turlari, tuzilishi, ishlatilish ko'lami, algoritmlar va ularni muayyan sharoitlarga mos holda tanlash usullari bo'yicha **yo'nalish** profiliga mos bilimlar darajasi bilan ta'minlashdir.

Fanning vazifasi – talabani ushbu fan bo'yicha olgan nazariy va Amaliy bilimlarini kurs loyihasi va bitiruv ishlarini bajarish bilan real sharoitga qo'llash bo'yicha ko'nikmalar hosil qilishdir.

Fan bo'yicha talabalarning tasavvur, bilim, ko'nikma va malakalariga qo'yiladigan talablar

«Ma'lumotlar tuzilmasi» o'quv fanini o'zlashtirish jarajonida amalga oshiriladigan masalalar doirasida bakalavr:

-Ma'lumotlar tuzilmalari va ularni qajta ishlash algoritmlari **haqida tasavvurga** ega bo'lishi;

- Ma'lumotlar tuzilmalari ustida amallarni bajarish;
- Ma'lumotlar tuzilmalari bilan ishlashda turli algoritmlardan foydalanishi;
- Algoritmlarni **tahlil** qilish va ularni yaratishni;
- Xotirani boshqarish usullarini bilishi va ulardan foydalana olishi;
- Xotirani boshqarishda turli Amaliy masalalarning dasturiy echimini

topish;

- Ma'lumotlar tuzilmalari va ular ustidagi amallardan foydalanish ko'nikmalariga ega bo'lshii kerak.

Fanning o'quv rejadagi boshqa fanlar bilan o'zaro bog'liqligi va uslubiy jihatdan uzviyligi

Ma'lumotlar tuzilmasi fani mutahassislik fanlar majmuasiga talluqli bo'lib, oliy o'quv yurtlarida o'qitiladigan oliy matematika, Informatika va axborot texnologiyalari, Hisoblash tizimlarida axborot asoslari, dasturlash texnologiyalari fanlari bilan uzviy bog'langan.

Fanning ilm-fan va ishlab chiqarishdagi o'rni

Elektron hisoblash mashinalariishi asosini ma'lumotlar tashkil etadi. Shuning uchun ham ma'lumotlarning tuzilishi, ularnig ishlash printsiplarini o'rganishni taqozo etadi. Ushbu umum kasbiy ilmiy fani hisoblani, ishlab chiqarish texnologik tizimining ajralmas bo'g'inidir.

Fanni o'qitishda zamonaviy axborot va pedagogik texnologiyalar

Kompyuter tarmoqlari fanini o'zlashtirishda masofadan o'qitish, o'quv qo'llanmalar va ma'ruzalar matnlarining elektron versiyalaridan, tarqatiladigan raqamli medialar, media to'plamlari ma'lumotlar bazasidan, proekcion apparatlardan, elektron plakatlar va virtual laboratoriya ishlaridan foydalanish ko'zda tutilgan.

Talabalar tomonidan bilim olish, bilimlarni o'zlashtirish va talabalarning bilimlarini haqqoniy baholash jarajonlarini axborot kommunikatsiya texnologiyalari vositalari yordamida o'tkazilishi ko'zda tutilgan. Ushbu vositalar yordamida "Kompyuter tarmoqlari"ning barcha bo'limlarini o'qitishda mul'timedia ma'ruza darslaridan foydalanish maqsadga muvofiqdir.

Ma'ruza va amaliy mashg'ulotlarda turli metod va vositalardan, hususan, aqliy hujum, klaster, amaliy ish va didaktik o'yinlar, portfolio, key-stadi, shuningdek,

Kompyuter dasturlaridan, internet tizimlaridan foydalanish mumkin.

ASOSIY QISM

Fanning nazariy mashg'ulotlari mazmuni

Algoritmlarni yaratish va tahlil qilish

Masaladan dasturga O'tish; ma'lumotlar turlari, ma'lumotlar tarkibi va ma'lumotlarning abstrakt turlari (tiplari); dasturlarni bajarilishi va bajarilish vaqtini **hisoblash**.

Ma'lumotlarning asosiy abstrakt turlari

“Ro'yxat” turdagi ma'lumotlarning abstrakt turlari va ro'yxatlarni realizatsiya qilish; stek, navbat, akslantirish; steklar va rekursiv proceduralar.

Daraxtlar

Asosiy tushunchalar; TREE ko'rinishdagi ma'lumotlarni abstrakt turlari; **daraxtlarni** realizatsiya qilish; ikkilik **daraxtlari**.

Yo'nalishi aniqlangan graflar

Asosiy tushunchalar; **yo'nalishi** aniqlangan **daraxtlarni** tasvirlash; eng qisqa yo'lni topish masalasi; juftlik qirralar orasidagi eng qisqa yo'lni topish; **yo'nalishi** aniqlangan graflarni aylanib o'tish, **yo'nalishi** aniqlangan ciklik graflar.

Yo'nalishi aniqlanmagan graflar

Asosiy tushunchalar; minimal narhli **daraxtlar** skeleti; **yo'nalishi** aniqlanmagan **daraxtlarni** aylanib t'tish; **daraxtlarni** juftli birlashtirish.

Saralash

Asosiy tushunchalar; Ichki saralash modeli; saralashning sodda sxemalari; tez saralash; “Cho'ntak” saralashi; taqqoslanma saralashlarni bajarilish vaqtlari; tartibli statistika.

Izlash

Asosiy tushunchalar; Ketma-ket izlash. Izlashning tezlashtirilgan usullari. Ikkilangan daraxt bo'yicha izlash. Ma'lumotlardan bevosita erkin foydalanadigan izlash usuli. Ko'p aspektli izlashning o'ziga xos xususiyatlari.

Amaliy mashg'ulotlarning taxminiy ro'yxati

1. Algoritmni yaratish va **tahlil** qilish.
2. Ma'lumotlarning **asosiy** abstrakt turlari.
3. **Daraxtlar**, ikkilik **daraxtlari**.
4. To'plamlarning **asosiy** operatorlarini realizatsiya qilish .
5. To'plamlarni tasvirlashni mahsus usullari MERGE va SPLIT operatorlari to'plami.
6. **Yo'nalishi** aniqlangan va **yo'nalishi** aniqlanmagan graflar.
7. Saralash: ichki saralash modeli.
8. Algoritmni **tahlil** etish algoritmlari va algoritmni yaratish usullari.
9. Tashqi Xotira uchun ma'lumotlar tarkibi va algoritmlar.

Amaliy mashg'ulotlarni tashkil etish bo'yicha tavsiyalar

Amaliy mashg'ulotlarda talabalar algoritmni tuzish, ma'lumotlarni tasvirlash usullari, algoritmni tahlil qilish va yaratish usullari asoslarini o'rganadilar.

Amaliy mashg'ulotlarni tashkil etish bo'yicha kafedra professor-O'qituvchilari tomonidan ko'rsatma va tavsiyalar ishlab chiqiladi. Unda talabalar **asosiy** ma'ruza mavzulari bo'yicha olgan bilim va ko'nikmalarini Amaliy masalalar yechish orqali yanada boyitadilar. Shuningdek, darslik va o'quv qo'llanmalar asosida talabalar bilimlarini mustahkamlashga erishish, tarqatma materiallardan foydalanish, ilmiy maqolalar va tezislarni chop etish orqali talabalar bilimni oshirish, masalalar yechish, mavzular bo'yicha ko'rgazmali qurollar tayyorlash va boshqalar tavsiya etiladi.

Laboratoriya mashg'ulotlarning taxminiy ro'yxati

1. Turli masalalar uchun algoritmlarni yaratish va **tahlil** qilish.
2. Chiziqli ro'yxatlar.
3. Stek..
4. Navbat.
5. Yozuv.
6. Ikkilik **daraxtlari**.
7. To'plamlardan foydalanib dasturlash. usullari
8. Saralash: ichki saralash modeli "Cho'ntak" saralashi.
9. Tanlash bilan saralash. Ko'pik saralash. To'g'ri kiritish bilan saralash.
10. Rekursiya.

Laboratoriya mashg'ulotlarni tashkil etish bo'yicha tavsiyalar

Laboratoriya ishlari talabalarda turli hil jarajonlarini **hisoblash**, algoritmlarini ishlab chiqish va tahlil qilish, to'plamlarining **asosiy** turlarini ajratish graflardan foydalanish, Xotira modellaridan foydalanish bo'yicha Amaliy ko'nikma va malaka hosil qiladilar.

Laboratoriya ishlarining tavsiya etiladigan mavzulari:

Kurs ishini tashkil etish bo'yicha uslubiy ko'rsatmalar

O'quv rejada kurs ishlari rejalashtirilmagan.

Mustaqil ta'limning shakli va mazmuni

Talaba mustaqil ishni tayyorlashda muayyan fanning hususiyatlarini hisobga olgan holda quyidagi shakllardan foydalanish tavsiya etiladi:

- darslik va o'quv qo'llanmalar bo'yicha fan boblari va mavzularini o'rganish;
- tarqatma materiallar bo'yicha ma'ruzalar qismini o'zlashtirish;

- avtomatlashtirilgan o'rgatuvchi va nazorat qiluvchi tizimlar bilan ishlash;
- mahsus adabiyotlar bo'yicha fanlar bo'limlari yoki mavzulari ustida ishlash;
- yangi texnikalarni, apparaturalarni, jarajonlar va texnologiyalarni o'rganish;
- talabaning o'quv-ilmiy-tadqiqot ishlarini bajarish bilan bog'liq bo'lgan fanlar bo'limlari va mavzularni chuqur o'rganish;
- faol va muammoli o'qitish uslubidan foydalaniladigan o'quv mashg'ulotlari;
- masofaviy (distansion) ta'lim.

Tavsiya etilayotgan mustaqil ishlarning mavzulari

1. Ma'lumotlar tarkibi va ma'lumotlarning abstrakt turlari (tiplari);
2. Steklar va rekursiv proseduralar
3. **Yo'nalishi** aniqlangan ciklik graflar
4. Taqqoslanma saralashlarni bajarilish vaqtlari
5. Lokal qidiruv algoritmlari
6. Har hil o'lchamli ob'ektlar uchun Xotira ajratish

Dasturning informatsion-uslubiy ta'minoti

Mazkur fanni o'qitish jarajonida ta'limning zamonaviy ilg'or interfaol usullaridan, pedagogik va axborot – kommunikaciya texnologiyalarining prezentaciya (taqdimot), multimedia va elektron-didaktik texnologiyalardan foydalaniladi. Amaliy mashg'ulotlarda aqliy hujum, klaster, blic-so'rov, guruh bilan ishlash, insert, taqdimot, kejs stadi kabi usul va texnikalardan keng foydalaniladi.

Foydalaniladigan adabiyotlar ro'yxati

Asosiy adabiyotlar:

1. Akbaraliev B.B. "Ma'lumotlar tuzilmasia algoritmlar" fanidan ma'ruzalar matni, Tashkent, 2008.
2. Akbaraliev B.B., Ysupova Z. Dj. «Ma'lumotlar tuzilmasi va algoritmlar»

fanidan laboratoriya ishlarini bajarish uchun uslubiy ko'rsatma. Tashkent, 2013 y.

3. Al'fred V. Aho, Djon Hopkroft, Deffri D. Ul'man. Strukturf dann'x i algoritmy//Ucheb.pos., M.: Izdatel'skiy dom "Vil'jame", 2000. – 384 s.

4. Robert Sedjvik. Fundamental'nye algoritmy na S++. Analiz, Struktury dann'x, Sortirovka. Poisk.//K.: Izd. «DiaSoft», 2001. – 688 s.

Qo'shimcha adabiyotlar:

1. Hudojberdiev M.H., Akbaraliev B.B. «Ma'lumotlar tuzilmasi va algoritmlar» fanidan Amaliy mashg'ulotlar uchun topshiriqlar. Toshkent, 2013.

2. Dinman M.I. S++. Osvoj na primerah//SPB.: BHV-Peterurg, 2006. – 384 s.

3. Berkell Djulian M. Fundamental'nye algoritme i struktury dann'x v Delphi//SPb.: OOO «DiaSoftJUP», 2003. – 560 s.

4. Virt N. Algoritmy i struktury dann'x. M.: Oberon, 2010.

Internet saytlar:

1. <http://www.edu-navigator.ru>

2. <http://www.phis.org.ru/informatika/>

3. www.pedagog.uz

4. www.Ziyonet.uz

5. www.edu.Uz

Foydalaniladigan adabiyotlar ro'yxati

Asosiy adabiyotlar:

1. Akbaraliev B.B. “Ma'lumotlar tuzilmasia algoritmlar” fanidan ma'ruzalar matni, Tashkent, 2008.
2. Akbaraliev B.B., Ysupova Z. Dj. «Ma'lumotlar tuzilmasi va algoritmlar» fanidan laboratoriya ishlarini bajarish uchun uslubiy ko'rsatma. Tashkent, 2013 y.
3. Al'fred V. Aho, Djon Hopkroft, Deffri D. Ul'man. Strukturuf dann'x i algoritmy//Ucheb.pos., M.: Izdatel'skiy dom "Vil'jame", 2000. – 384 s.
4. Robert Sedjvik. Fundamental'nye algoritmy na S++. Analiz, Struktury dann'x, Sortirovka. Poisk.//K.: Izd. «DiaSoft», 2001. – 688 s.

Qo'shimcha adabiyotlar:

1. SH. Mirziyoyev Buyuk kelajagimizni mard va olijanob xalqimiz bilan birga quramiz. Toshkent “O'zbekiston” 2017. 488 b.
2. SH. Mirziyoyev. Qonun ustuvorligi va inson manfaatlarini ta'minlash – yurt taraqqiyoti va xalq farovonligining garovi. O'zbekiston Respublikasi Konstitutsiyasi qabul qilinganligining 24 yilligiga bag'ishlangan tantanali marosimdagi ma'ruza. 2016-yil 7-dekabr. Toshkent - “O'zbekiston” - 2017. 32 b.
3. Mirziyoev SH.M. Tanqidiy tahlil, qat'iy tartib-intizom va shaxsiy javobgarlik – har bir rahbar faoliyatining kundalik qoidasi bo'lishi kerak. O'zbekiston Respublikasi Vazirlar Mahkamasining 2016 yil yakunlari va 2017 yil istiqbollarga bag'ishlangan majlisdagi O'zbekiston Respublikasi Prezidentining nutqi. // Xalq so'zi gazetasi. 2017 yil 16 yanvar, №11.
4. SH.Mirziyoev. Erkin va farovon, demokratik O'zbekiston davlatini birgalikda barpo etamiz. O'zbekiston Respublikasi Prezidenti lavozimiga kirishish tantanali marosimiga bag'ishlangan Oliy Majlis palatalarining qo'shma majlisidagi nutqi. Toshkent – “O'zbekiston”. 2016. 56 b.

5. O'zR «Axborotlashtirish to'g'risida» 2003 yil 11 dekabrda Qonuni.
6. Bjarne Stroustrup. The C++ Programming Language (4th Edition). Addison-Wesley, 2013. 1363 page.
7. Hudoberdiev M.H., Akbaraliev B.B. «Ma'lumotlar tuzilmasi va algoritmlar» fanidan Amaliy mashg'ulotlar uchun topshiriqlar. Toshkent, 2013.
8. Dinman M.I. S++. Osvoj na primerah//SPB.: BHV-Peterurg, 2006. – 384 s.
9. Berkell Djulian M. Fundamental'nye algoritme i struktury dann'x v Delphi//SPb.: OOO «DiaSoftJUP», 2003. – 560 s.
10. Virt N. Algoritmy i struktury dann'x. M.: Oberon, 2010.

Elektron manbalar

1. <http://cppstudio.com> – C++ tilida programmalash bo'yicha namunalar izoxlari bilan keltirilgan
2. <http://cplusplus.com> – C++ tilida mavjud konstruktsiyalar ta'rifi, ishlatish namunalari bilan keltirilgan.
3. <http://www.compteacher.ru/programming> – dasturlash bo'yicha video darsliklar mavjud.
4. <http://www.intuit.ru> – internet universitet, dasturlash bo'yicha yozma va video ma'ruzalar o'qish, test sinovlaridan o'tish va sertifikat olish imkoniyati mavjud.
5. <http://www.ziyonet.uz> – dasturlash asoslari bo'yicha referatlar topish mumkin.
6. <http://www.edu-navigator.ru>
7. <http://www.phis.org.ru/informatika/>
8. www.pedagog.uz
9. www.Ziyonet.uz