

Тошкент Ахборот Технологиялари Университети



Курс ИШИ

Мавзу: Динамик структуралар асосида баланс дарахтини
қуриш.

Бажарди: Алимов А.

Текширди:

Toshkent 2013

Режа

Кириш

I Тахлили кисм

- 1.1 Дарахтлар хакида тушунча
- 1.2 Бинар дарахтлар
- 1.3 Структуралар
- 1.4 Массивлар хакида тушунча
- 1.5 Бир улчовли массивлар.

II Асосий кисм.

- 2.1. Масаланинг кўйилиши.
- 2.2. Масаланинг алгоритми.
- 2.3. Дастур коди.
- 2.4. Олинган натижанинг тахлили

Хулоса.

Фойдаланилган адабиётлар рўйхати.

Илова.

Мундарижа

1.	Кириш.....	4
2.	Тахлили кисм	5
3.	1.1 Дарахтлар хакида тушунч.....	5
4.	1.2 Баланс дарахтлар.....	6
5	1.3 Структуралар.....	7
6	1.4 Массивлар хакида тушунча.....	9
7	1.5 Бир улчовли массивлар.....	10
8	Асосий кисм.....	12
9	2.1. Масаланинг кўйилиши.....	12
10	2.2. Масаланинг алгоритми.....	12
11	2.3.Олинган натижанинг тахлили.....	13
12	Хулоса.....	14
13	Фойдаланилган адабиётлар рўйхати.....	15
14	Илова.....	16

Кириш

Хозирги кунда оламни яни бу дунёни техникасиз тасаввур килиб бўлмайди. Хаётимизга шу кадар кириб келдики, биз унга жуда хам боғланиб қолдик. Мисол килиб олсак оддий нарса мабил телефонимиз уни бир кун ишлатмасак ёки бирор жойда унитиб қолдирса ўзимизни бошқача тутамиз.

Хозирги аср техника асри ҳисобланар экан хоҳлаймизми йўқми унга дастур ёзиш керак бўлади. Дастурчи дастур ёзишидан олдин қанақа техника ёки офис учунми ёки ҳаммасини сўраб суриштириб ундан кейин ишни бошлайди. Шунини тан олишимиз керакки нима бўлсаям техника олдин ривожланиб кейинчалик унга дастур ёзилади.

Дастур ёзиш учун бир нечта тиллар мавжуд ҳисобланади. Булар жумласига C#, C++, Java, Pascal, Delphi ва бошқа тилларни санаб ўтишимиз мумкин.

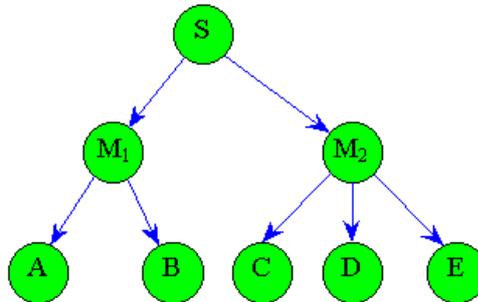
Биз талабаларга ҳам шу ишларни яни дастурий таъминотни дастурлаш тилларини муқаммал ўрганишда, шу пайтгача олган билимларимизни шу курс ишимизда кўрсатамиз ва таҳлилқиламиз.

Бу курс ишимни ёзишдан мақсад ҳозирги пайтда шу кадар маълумотлар кўпайиб кетдики биз уни тасаббуригимизга сизгидира олмаймиз.

Маълумотларни имкон қадар тезроқ топишда, ёки вақтни йўқотмасликка ҳаракат қиламиз. Биз қидираётган маълумотларни саралаб топишда жуям аҳамияти катта ҳисобланади. Бу қилган курс ишимда Мисол килиб оладиган бўлсак мабил телефонда гаплашган вақтларни ёки бўлмаса оддий килиб айтсак гаплашган вақтлар ичидан энг кичигини ёки энг кўп гаплашгани аниқлашда жуда ката ва тез ҳамда ишончли топиб беради ва бир тамонда кичик иккинчи тамонда эса катталарни чиқариб беради.

Дарахтлар

Дарахт – бу чизиксиз боғланган маълумотлар тузилмасидир (қаранг, чизма).



Дарахт ўзининг қуйидаги белгилари билан таснифланади:

- дарахтда шундай битта элемент борки, унга бошқа элементлардан муурожаат йўқ. Мазкур элементга дарахт илдизи дейилади;
- дарахтда ихтиёрий элементга чекли сондаги кўрсаткичлар ёрдамида муурожаат қилиш мумкин;
- дарахтнинг ҳар бир элементи фақатгина ўзидан олдинги келган битта элемент билан боғланган. Дарахтнинг ҳар бир тугуни оралиқ ёки терминал (барг) бўлиши мумкин. Юқоридаги чизмада M_1 , M_2 - оралиқ, A , B , C , D , E - барглاردир. Терминал тугуннинг ўзига хос таснифи унинг шохлари йўқлигидир.

Баландлик – бу дарахт босқичи сони. Юқоридаги чизмадаги дарахт баландлиги иккига тенг.

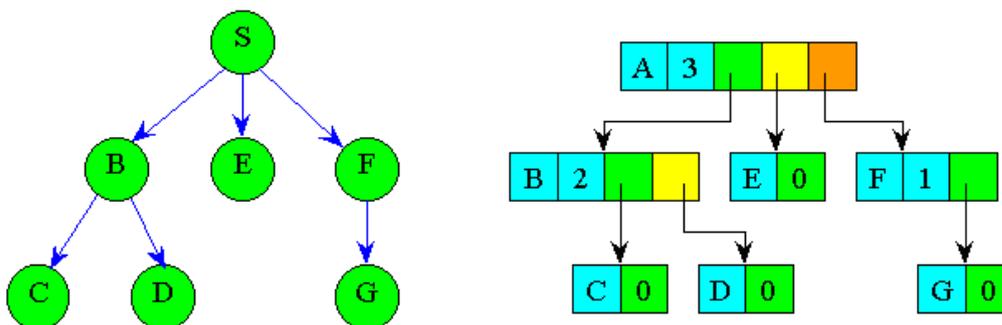
Дарахт тугунларидан чиқаётган шохлар сони тугундан чиқиш даражаси дейилади (Келтирилган чизмада M_1 учун чиқиш даражаси 2, M_2 учун эса 3 га тенг). Дарахтлар чиқиш даражаси бўйича синфларга ажратилади:

- 1) агар максимал чиқиш даражаси m бўлса, у ҳолда бундай дарахт m -чи тартибли дарахт дейилади;
- 2) агар чиқиш даражаси 0 ёки m бўлса, у ҳолда тўлиқ m -чи тартибли дарахт бўлади;

3) агар максимал чиқиш даражаси 2 бўлса, у ҳолда бундай дарахт бинар дарахт дейилади;

4) агар чиқиш даражаси 0 ёки 2 бўлса, у ҳолда тўлиқ бинар дарахт дейилади. уғунлар орасидаги боғлиқликни тавсифлаш учун яна қуйидагича терминдан фойдаланилади: M_1 – A ва B элементлар учун “ота”. A ва B – эса M_1 тугун “ўғиллари”.

Дарахтларни тасвирлаш



Дарахтни график шаклдаги ва унинг чизиксиз рўйхат шаклидаги ифодаланиши

ЭХМ хотирасида дарахтни ифодалашнинг энг қулай усули бу уни боғланган рўйхатлар кўринишида ифодалашдир. Рўйхат элементи тугун қиймати ва чиқиш даражасини ўз ичига олувчи информацион майдонга ҳамда чиқиш даражасига тенг бўлган кўрсаткичлар майдонига эга бўлиши лозим (юқоридаи чизма), яъни элементнинг ҳар бир кўрсаткичи ушбу элементни тугун ўғиллари бўлган тугунларга йўналишини аниқлайди.

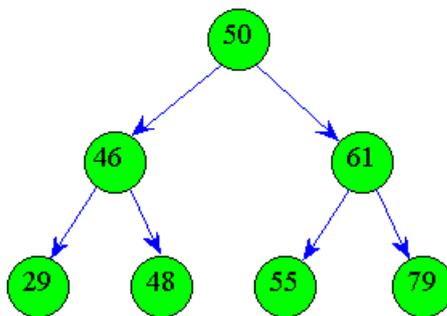
Баланс дарахтлар

Бинар дарахтлар энг кўп фойдаланиладиган дарахтлар тури ҳисобланади.

Дарахтларни ЭХМ хотирасида тасвирланишига кўра ҳар бир элемент тўртта майдонга эга ёзув ҳисобланади. Мазкур майдонлар қиймати мос равишда ёзув калити бўлиб, бошқа элементларга мурожаатни ифодалайди, яъни чапга-пастга, ўнга-пастга ва ёзув матнига.

Шуни эсда тутиш лозимки, дарахт ҳосил қилинаётганда, отага нисбатан чап томондаги ўғил қиймати кичик калитга, ўнг томондаги ўғил эса катта

қийматли калитга эга бўлади. Масалан, қуйидаги элементлардан баланс дарахт қурамиз: 50, 46, 61, 48, 29, 55, 79. У қуйидаги кўринишга эга бўлади:



Натижада, ўнг ва чап қисм дарахтлари бир хил босқичли тартибланган баланс дарахт ҳосил қилдик. Агар дарахтнинг ўнг ва чап қисм дарахтлари босқичлари фарқи бирдан кичик бўлса, бундай дарахт идеал мувозанатланган дарахт дейилади. Юқорида ҳосил қилган бинар дарахтимиз идеал мувозанатланган дарахтга мисол бўлади.

Баланс дарахтни ҳосил қилиш учун ЭХМ хотирасида элементлар қуйидаги турда бўлиши лозим:



$V = \text{MakeTree}(\text{Key}, \text{Rec})$ амали иккита кўрсаткичли (калит) ва иккита майдонли (информацион) элемент яратади (дарахт тугуни)

Бошида калит биринчи қиймати киритилади. Ундан сўнг элементни ўзини maketree процедураси орқали ҳосил қиламиз. Кейин эса кўрсаткич бўш қийматни кўрсатгунча циклни давом эттираемиз.

Структуралар

Структура – бу маълумотларни бир бутун номланган элементлар тўпламига бирлаштириш. Структура элементлари (майдонлар) ҳар хил типда бўлиши мумкин ва улар ҳар хил номларга эга бўлиши керак.

Структурали тип қуйидагича аниқланади:

```
struct { <таърифлар рўйхати > }
```

Структурада албатта битта компонента бўлиши керак. Структура типдаги ўзгарувчи қуйидагича таърифланади:

```
<структура_номи > <ўзгарувчи>;
```

Структура типдаги ўзгарувчи таърифланганда инициализация қилиниши мумкин:

```
<структура_номи > <ўзгарувчи>=<инициализатор>;
```

Структурани инициализациялаш учун унинг элементлар қийматларини фигурали қавсларда тавсифланади.

Мисоллар:

1. struct Student

```
{  
    char name[20];
```

```
    int kurs;
```

```
    float rating;
```

```
};
```

```
Student s={"Қурбонов",1,3.5};
```

2. struct

```
{  
    char name[20];
```

```
    char title[30];
```

```
    float rate;
```

```
}employee={"Ашуров", "директор",10000};
```

Структураларни ўзлаштириш. Битта тузилма типдаги ўзгарувчилар учун ўзлаштириш операцияси аниқланган. Бунда ҳар бир элементдан нусха олинади. Масалан:

```
Student ss=s;
```

Структура элементларига мурожаат. Структура элементларига мурожаат аниқланган исмлар ёрдамида бажарилади:

<Структура_номи>.<элемент_номи>

Примеры:

employee.name – «Ашуров» сатрига кўрсаткич;

employee.rate – 10000 қийматга эга бўлган бутун типдаги ўзгарувчи

Структурага кўрсаткичлар. Структурага кўрсаткичлар оддий кўрсаткичлар каби тасвирланади:

Student*ps;

Структурага кўрсаткич таърифланганда инициализация килиниши мумкин:

Student *ps=&mas[0];

Кўрсаткич оркали структура элементларига икки усулда мурожаат килиш мумкин. Биринчи усул адрес буйича қиймат олиш амалига асосланган бўлиб қуйидаги шаклда қўлланилади:

(* структурага кўрсаткич).элемент номи;

Иккинчи усул махсус стрелка (->) амалига асосланган бўлиб қуйидаги кўринишга эга:

структурага кўрсаткич->элемент номи

Структура элементларига қуйидаги мурожаатлар узаро тенгдир:

cin>>>(*ps).name;

cin>>ps->title;

МАССИВЛАР

Бу қисмда дастурдаги маълумот структуралари билан танишишни бошлаймиз.

Дастурда икки асосий тур маълумот структуралари мавжуддир. Биринчиси статик, иккинчиси динамикдир. Статик деганимизда хотирада эгаллаган жойи

Ўзгармас, дастур бошида бериладиган структураларни назарда тутамиз. Динамик маълумот типлари дастур давомида ўз хажмини, эгаллаган хотирасини ўзгартириши мумкин.

Агар структура бир хил катталиқдаги типлардан тузилган бўлса, унинг номи массив (array) дейилади. Массивлар дастурлашда энг кўп қўланиладиган маълумот типларидир. Бундан ташқари структуралар бир неча фарқли типдаги ўзгарувчилардан ташкил топган бўлиши мумкин. Буни биз клас (Пасцалда record, C++ класс) деймиз. Масалан бундай структурамиз ичида одам исми ва ёши бўлиши мумкин.

Бу бўлимда биз массивлар билан яқиндан танишиб ўтамиз. Бу бўлимдаги массивларимиз C услубидаги, поинтерларга (кўрсаткичларга) асослан структуралардир. Массивларнинг бошқа кўринишларини кейинги қисмларда ўтамиз.

Массивлар хотирада кетма-кет жойлашган, бир типдаги ўзгарувчилар гуруҳидир.

Алоҳида бир ўзгарувчини кўрсатиш учун массив номи ва керакли ўзгарувчи индексини ёзамиз. C/C++ даги массивлардаги элементлар индекси ҳар доим нолдан бошланади. Бизда ҳар типдаги m номли массив бор бўлсин.

Бир улчовли массивлар.

Массив бу бир типли номерланган маълумотлар жамланмасидир. Массив индексли ўзгарувчи тушунчасига мос келади. Массив таърифланганда тип, номи ва индекслар чегараси кўрсатилади. Мисол учун лонг `int a[5]; char w[200]; double f[4][5][7]; char[7][200]`. Массив индекслар ҳар доим 0 дан

бошланади. C ++ тили стандарти бўйича индекслар сони 31 тагача бўлиши мумкин, лекин амалда бир ўлчовли ва икки ўлчовли массивлар қўлланилади. Бир ўлчовли массивларга математикада вектор тушунчаси мос келади. Массивнинг `int z[3]` шаклдаги таърифи, `int` типига тегишли `z[0],z[1],z[2]` элементлардан иборат массивни аниқлайди.

Массивлар таърифланганда инициализация қилиниши, яъни бошлангич қийматларлари кўрсатилиши мумкин. Мисол учун:

```
float c[]={1,-1,2,10,-12.5};
```

Бу мисолда массив чегараси автоматик аниқланади. Агар массив инициализация қилинганда элементлар чегараси кўрсатилган бўлса , руйхатдаги элементлар сони бу чегарадан кам бўлиши мумкин, лекин ортиқ бўлиши мумкин эмас. Мисол учун `int A[5]={2,-2}`. Бу ҳолда `a[0]` ва `a[1]` қийматлари аниқланган бўлиб, мос ҳолда 2 ва -2 га тенг.

Масаланинг кўйлиши

Берилган масалани яни Динамик структуралар асосида баланс дарахитини куриш.

Дастур алгаритими

Ман ўзимга берилган яни вариантимдаги масалани ечишда C++ ни танлади ва шу тили келажакда янада мукамал килиб ўлаштирмоқчиман.

Биринчи навбатда бизга дастур кодини киритиш талаб этилади. Бунинг икки хил усули бор, фойдаланувчи ўзи киритади (дастур кодини ўзи ёзиб чиқади) ёки дастур коди ёзилган файлни юклаб олади. Дастур коди киритилгач, яни F9 ни боссак тугмаси босилади ва тахлил бошланади .

Тахлил деганда нимани тушунамиз ва у кандай амалга оширилади? Тахлил деганда (бизнинг дастурий таъминот учун), киритилган дастур матнида берилган изоҳли каторни топиш ва уларни ўчиришни тушунамиз.

Олинган натижанинг тахлили ёки дарахт чизиш ойнаси

Binar daraxti

Elementlar sonini kirting:

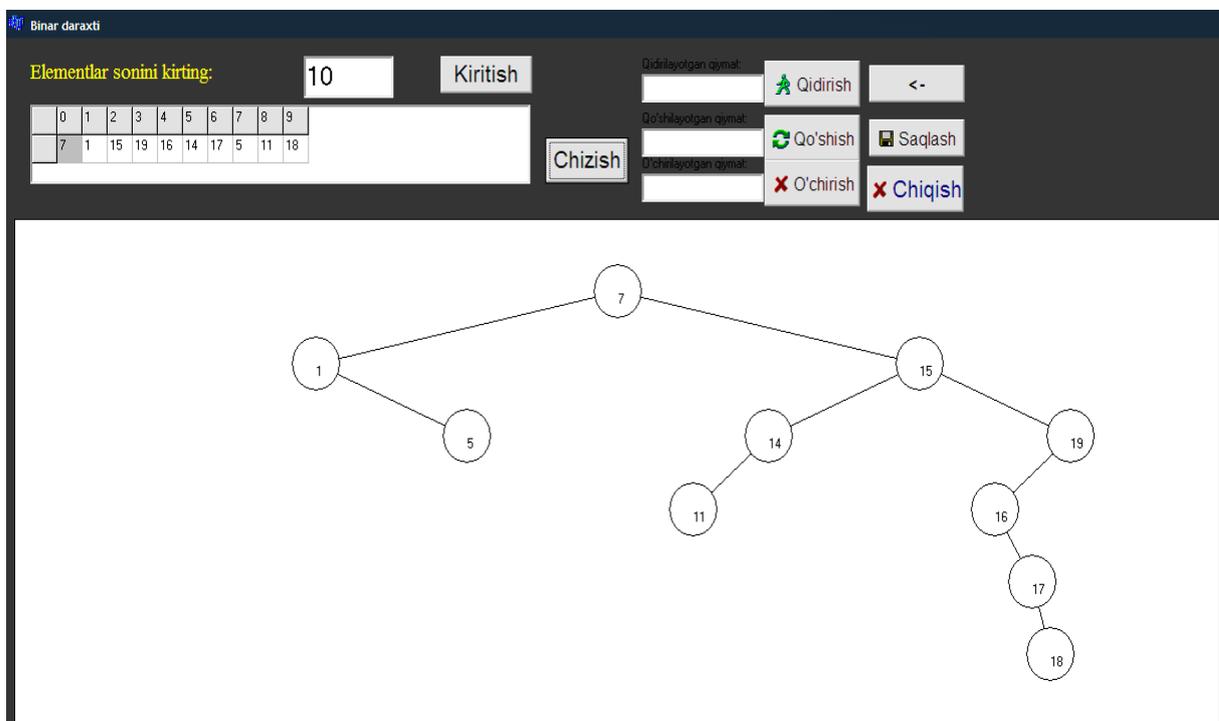
Qidirilayotgan qiymat:

Qo'shilayotgan qiymat:

O'chirilayotgan qiymat:

Мана биз уч керак асосий жойга келдик.

Биз кўриб турган ойнада ўзимизнинг баланс дарахтимизни курамыз, бунинг учун биз белгиланган жойга сон киритишимиз керак бўлади.



Хулоса.

Хулосам шуки курс ишимни килишда яни дарахтлар, динамик структураларнинг абзаллики шуки хаётимизда фойдаланадига кўпгина нарсалар яни саралаш ҳам учраб туради. Хар бир инсон ўзига кулай нарсани ёки кулайлик яратишни хохлайди. Масалан мабил алокадан фойдаланамиз унда биз гаплашга вақтлар ёки туша олмаган ракамлар хакида осон малумотни топишда бизга жуда ҳам кулай хисобланади.

Бу курс ишимни килишдан мақсад хар бир талабанинг ўзининг билимини янада мустахкамлайди

Бу курс иши менга замонавий дастурлаш тиллари структурасини ўрганишда жуда катта ёрдам берди. Бу ўрганган билимларим келажакда менга ёрдам беришига ишонаман.

Фойдаланилган адабиётлар рўйхати.

1. *Алфред В. Ахо., Джон Э. Хопкрофт, Джеффри Д. Ульман. Структура данных и алгоритмы//Учеб.пос., М. : Изд.дом: "Вильямс", 2000, — 384 с.*
2. *Бакнелл Джулиан М. Фундаментальные алгоритмы и структуры данных в Delphi//СПб: ООО «ДиаСофтЮП», 2003. 560с.*
3. *Роберт Седжвик. Фундаментальные алгоритмы на С++. Анализ, Структуры данных, Сортировка, Поиск//К.: Изд. «ДиаСофт», 2001.- 688 с.*
4. *Динман М.И. С++. Освой на примерах//СПБ.:БХВ-Петербург, 2006, 384.*
5. *Шилдт, Герберт. Полный справочник по С#//М. : Изд. дом "Вильямс", 2004, 752 с.*
6. *Вирт Н. Алгоритмы и структуры программы//М., Мир, 1985.*
- 7 www.ziyonet.uz.
- 8 www.tuit.uz

Илова

Dastur kodi

```
//-----  
  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Unit1.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
class node  
{  
    public:  
        int info;  
        node *left;  
        node *right;  
        int nx;  
};
```

```

int n,key1=-12221;
int a[100];
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    srand(time(0));
    n=StrToInt(Edit1->Text);
    StringGrid1->ColCount=n+1;
    int maxrand=20;
    if(n>maxrand) maxrand=n+3;
    for(int i=0;i<3000;i++)
        StringGrid1->Cells[i+1][0]=i;
    ProgressBar1->Max=n; ProgressBar1->Visible=True;
    StringGrid1->Cells[1][1]=rand()%5+maxrand/2-3;
    for(int i=1;i<n;i++)
    {
        C1:
        int kk=0;
        StringGrid1->Cells[i+1][1]=rand()%maxrand;
        for(int j=0;j<i;j++)
            if(StringGrid1->Cells[i+1][1]==StringGrid1->Cells[j+1][1])
                {kk++;break;}
        if(kk!=0) goto C1;
        ProgressBar1->Position++;
    }
    StringGrid1->Visible=true;
    Button2->Enabled=true;
    ProgressBar1->Visible=False;
}
//-----

```

```

int ss[300],sch=0;
void __fastcall TForm1::Button2Click(TObject *Sender)
{
    if(key1==0)
        key1=StrToInt(LabeledEdit1->Text);
        node *tree=NULL;
        node *next=NULL;
        int s,qw=0;
        int a[300],j,max=0;
        for(j=0;j<300;j++) a[j]=0;j=0;
        n=StrToInt(Edit1->Text);
        StringGrid1->ColCount=n+1;

        for(int i=0;i<n;i++)
        {
            node *p=new node;
            node *last=new node;
            s=StrToInt(StringGrid1->Cells[i+1][1]);
            int kk=0;
            for(int ii=0;ii<i;ii++)
            {
                if(ss[ii]==ss[i]) kk=0;
            }
            if(kk>0) continue;
            p->info=s; ss[i]=s;
            p->left=NULL;
            p->right=NULL;
            if(i==0){ tree=p; next=tree; continue;}
            next=tree;

```

```

        j=0;
        while(1){ j++;
            last=next;
            if(p->info<next->info) next=next->left; else next=next-
>right;

            if(next==NULL) break;
        }
        if(p->info<last->info)last->left=p; else last->right=p;
3    a[j]++;
        if(max<j) max=j;
    }
    int maxj[100],b[100];
    int wh=Image1->Width,ht=Image1->Height;
    Image1->Canvas->Brush->Color=clWhite;
    Image1->Canvas->Rectangle(0,0,wh,ht);
    Image1->Canvas->Brush->Color=clWhite;

    b[0]=wh/2;
    for(j=1;j<=max;j++)
    {
        b[j]=b[j-1]/2;
    }
    int x1,y1,x,y,R;
    if(max!=0) R=wh/(10*max);else R=wh/10;
    node *tree1=NULL;
    node *next1=NULL;
    for(int i=0;i<n;i++)
    {
        node *p=new node;

```

```

node *last=new node;
s=StrToInt(StringGrid1->Cells[i+1][1]);
int kk=0;
for(int ii=0;ii<i;ii++)
{
    if(ss[ii]==ss[i]) kk++;
}
if(kk>0) continue;
p->info=s; ss[i]=s;
p->left=NULL;
p->right=NULL;
p->nx=0;
if(i==0){
x=wh/2; y=ht/(max+2);
if(p->info==key1)
{
    Image1->Canvas->Pen->Color=clRed;
    Image1->Canvas->Brush->Color=clAqua;
    qw=1;
}
Image1->Canvas->Ellipse(x-R,y-R,R+x,R+y);
Image1->Canvas->TextOutA(x,y,IntToStr(p->info));
Image1->Canvas->Pen->Color=clBlack;
Image1->Canvas->Brush->Color=clWhite;
p->nx=b[0];
tree1=p; next1=tree1; last->nx=x; continue;}
next1=tree1;
j=0;
while(1){ j++;

```

```

        last=next1;
        if(p->info<next1->info)
        {
            next1=next1->left;
        }
        else
        if(p->info>next1->info)
        {
            next1=next1->right;
        }
        else break;
        if(next1==NULL) break;
    }
int rl=-1;
if(p->info<last->info)
{
    last->left=p;
    last->left->nx=last->nx-b[j];
    x=last->left->nx;
    x1=last->nx;
}
else
if(p->info>last->info)
{
    last->right=p;
    last->right->nx=last->nx+b[j];
    x=last->right->nx;
    x1=last->nx;
}
rl=1;

```

```

    }
    else break;
    y1=j*ht/(max+2);
    y=(j+1)*ht/(max+2);
    Image1->Canvas->MoveTo(x,y);
    Image1->Canvas->LineTo(x1,y1);
    Image1->Canvas->Brush->Color=clWhite;
    Image1->Canvas->Ellipse(x-R,y-R,R+x,R+y);
    Image1->Canvas->Ellipse(x1-R,y1-R,R+x1,R+y1);
    if(last->info==key1)
    {
        Image1->Canvas->Pen->Color=clRed;
        Image1->Canvas->Brush->Color=clAqua;
        qw=1;
    }
    Image1->Canvas->TextOutA(x1,y1,IntToStr(last->info));
    Image1->Canvas->Pen->Color=clBlack;
    Image1->Canvas->Brush->Color=clWhite;
    if(p->info==key1)
    {
        Image1->Canvas->Pen->Color=clRed;
        Image1->Canvas->Brush->Color=clAqua;
        qw=1;
    }
    Image1->Canvas->TextOutA(x,y,IntToStr(p->info));
    Image1->Canvas->Pen->Color=clBlack;
    Image1->Canvas->Brush->Color=clWhite;
}
if(key1!=-12221&&qw==0&&sch!=0)

```

```

        ShowMessage(IntToStr(key1)+" qiymat topilmadi!!!");
    }
//-----
void __fastcall TForm1::BitBtn1Click(TObject *Sender)
{
    exit(1);
}
//-----
void __fastcall TForm1::BitBtn2Click(TObject *Sender)
{
    key1=0; sch=1;
    Button2->Click(); sch=0;
}
//-----

void __fastcall TForm1::BitBtn3Click(TObject *Sender)
{
    if(LabeledEdit2->Text!=EmptyStr)
    {
        int kk=0;
        int news=StrToInt(LabeledEdit2->Text);
        for(int ii=1;ii<=n;ii++)
        {
            if(StrToInt(StringGrid1->Cells[ii][1])==news) kk++;
        }
        if(kk==0)
        {
            ShowMessage("Qo'shish muvoffiqiyatli amalga oshdi!!!");
        }
    }
}

```

```

        StringGrid1->ColCount++;n++;
        Edit1->Text=IntToStr(n);
        StringGrid1->Cells[StringGrid1->ColCount-1][1]=LabeledEdit2-
>Text;

        key1=-12221;
        Button2->Click();
    }
    else
        ShowMessage("Qo'shish amalga oshmadi. Siz kiritgan qiymat
daraxtda bor!");
    }
    else
        ShowMessage("Qo'shiladigan qiymatni kiriting!");
    }
//-----

String stringgrid[1000],txt1;
void __fastcall TForm1::BitBtn4Click(TObject *Sender)
{
if(LabeledEdit3->Text!=EmptyStr)
{
    int kk=n+1;
    int dels=StrToInt(LabeledEdit3->Text);
    for(int ii=1;ii<=n;ii++)
    {
        if(StrToInt(StringGrid1->Cells[ii][1])==dels) {kk=ii;break;}
    }
    if(kk<n+1)
    {

```

```

for(int jj=1;jj<=n;jj++)
    stringgrid[jj]=StringGrid1->Cells[jj][1];
txt1=Edit1->Text;
int min,kk1;
if (kk==n) { min=0;kk1=kk;}
    else
    {
        min=abs(StrToInt(StringGrid1->Cells[kk][1])-
StrToInt(StringGrid1->Cells[kk+1][1]));
        kk1=kk+1;
        for(int ii=kk+1;ii<n;ii++)
            if(abs(StrToInt(StringGrid1->Cells[kk][1])-StrToInt(StringGrid1-
>Cells[ii][1]))<min)
                { min=abs(StrToInt(StringGrid1->Cells[kk][1])-
StrToInt(StringGrid1->Cells[ii][1]));
                    kk1=ii;}
        StringGrid1->Cells[kk][1]=StringGrid1->Cells[kk1][1];
    }
for(int ii=kk1;ii<n;ii++)
    {
        StringGrid1->Cells[ii][1]=StringGrid1->Cells[ii+1][1];
    }
key1=-12221;
//ShowMessage("O'chirish muvoffiqiyatli amalga oshdi!!!");
n--;
StringGrid1->ColCount--;
Edit1->Text=IntToStr(n);
Button2->Click();
}

```

```

        else
            ShowMessage("O'chirish amalga oshmadi. O'chirilayotgan qiymat
daraxtda yo'q!");
    }
    else
        ShowMessage("O'chirilayotgan qiymatni kiriting!");
    }
//-----

void __fastcall TForm1::FormCreate(TObject *Sender)
{
    Image1->Width=1200;
}
//-----

void __fastcall TForm1::BitBtn5Click(TObject *Sender)
{
    if(SavePictureDialog1->Execute())
    {
        Image1->Picture->SaveToFile(SavePictureDialog1->FileName+".jpg");
    }
}
//-----

void __fastcall TForm1::BitBtn6Click(TObject *Sender)
{
    for(int jj=1;jj<=n;jj++)
        StringGrid1->Cells[jj][1]=stringgrid[jj];
    Edit1->Text=txt1;
}

```

```
n=StrToInt(txt1);  
Button2->Click();  
}  
//-----
```

Foydalanilgan adabiyotlar ro'yxati.

1. Alfred V. Aho., Djon E. Hopkroft, Djefri D. Ul'man. Struktura danih va algoritmi//Ucheb.pos., M. : Izd.dom: "Vil'yams", 2000, - 384 s.
2. Baknell Djulian M. Fundamental'nie algoritmi va strukturi danih v Delphi//SPb: OOO «DiaSoftYUP», 2003. 560s.
3. Robert Sedjvik. Fundamental'nie algoritmi na C++. Analiz, Strukturi danih, Sortirovka, Poisk//K.: Izd. «DiaSoft», 2001.- 688 s.
4. Dinman M.I. S++. Osvoyn na primerah//SPB.:BHV-Peterburg, 2006, 384.
5. Schildt, Gerbert. Polniy spravochnik po S#//M. : Izd. dom "Vil'yamc", 2004, 752 s.
6. Virt N. Algoritmi va strukturi programmi//M., Mir, 1985.

7. Maruzalar matni.

8. www.ziyonet.uz

www.google.com

www.tuit.uz