

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

ALISHER NAVOIY NOMIDAGI SAMARQAND DAVLAT UNIVERSITETI

**MEXANIKA-MATEMATIKA
FAKULTETI**

«MATEMATIK MODELLSHTIRISH» KAFEDRASI

**«EKSTREMAL MASALALARNI YECHISHDA GRAFLAR NAZARIYASI
USULLARIDAN FOYDALANISH JARAYONINI NAMOYISH QILISH»
mavzusidagi**

BITIRUV MALAKAVIY ISHI

Bajardi: «Amaliy matematika va
informatika» ta'lim yo'nalishi
bitiruvchi 4 kurs talabasi
OCHILOV KAMOLIDDIN
ABDURAZZOQOVICH.

Ilmiy rahbar:
dots. BOZOROV I.N.

Bitiruv malakaviy ishi kafedradan dastlabki himoyadan o'tdi.

_____ sonli bayonnomasi «_____» _____ 2013 yil.

SAMARQAND – 2013 y.

MUNDARIJA

Kirish.....	3
I BOB. Graflar nazariyasining asosiy tushunchalari.....	6
1.1. Graflar nazariyasining boshlang'ich ma'lumotlari.....	6
1.2. Graflarning berilish usullari.....	9
1.3. Graflar ustida amallar.....	16
1.4. Marshrutlar va zanjirlar.....	18
1.5. Eyler va Gamilton graflari.....	22
1.6. Grafning metrik xarakteristikalar.....	23
 II BOB. Minimal masofaga ega bo'lgan masalalar va ularni yechish algoritmлари hamda dasturiy vositalari.....	 27
2.1. Ba'zi ekstremal masalalarni yechishda graflar nazariyasi elementlaridan foydalanish (masalaning qo'yilishi va uni yechish algoritmi).....	27
2.2. Dasturdan foydalanuvchilar uchun ko'rsatmalar.....	30
 Xulosa.....	 34
Foydalanilgan adabiyotlar.....	36
Ilovalar	

Kirish

Masalaning qo'yilishi. Turli ekstremal masalalarni yechishda graflar nazariyasining tatbiqi sifatida Deyskra algoritmidan foydalanib, minimal masofaga ega bo'lgan yo'lni aniqlash hamda qiziqarli boshqotirma (kalkulyator) masalasi uchun ob'ektga yo'naltirilgan "Delphi" dasturlash tilida viziullashtirilgan dasturiy vositalarni ishlab chiqish.

Mavzuning dolzarbligi. Tabiatda ko'pincha, narsa va hodisalarning xossalari o'rganish jarayonida o'rganilayotgan ob'yekt elementlarini bir-birlari bilan taqqoslab, ularni birgalikda qarab yoki elementlarni bo'laklarga ajratib turli xulosalar qilishga to'g'ri keladi. **Graflar nazariyasi** boshqotirmalar va qiziqarli o'yinlarni o'rganish jarayonida paydo bo'lib, hozirgi vaqtda graf tushunchasi yordamida yo'llar, elektr, axborot va boshqa tarmoqlar, geografik xaritalar, kimyoviy birlashmalar, odamlar va jamiyat orasidagi munosabatlar bilan bog'liq hamda boshqa ko'plab masalalarni hal qilish mumkin. Graflar nazariyasi axborot texnologiyalar rivojida muhim ahamiyatga ega bo'lgan diskret matematikaning bir tilidir.

Shu sababli ushbu ishda ba'zi ekstremal masalalarni yechishda graflar nazariyasining tatbiqlari sifatida Deyskra algoritmidan foydalanib, minimal masofaga ega bo'lgan yo'lni aniqlash hamda qiziqarli boshqotirma (kalkulyator) masalasi uchun dasturiy vositalarini ishlab chiqish masalalari qaraladi.

Ishning maqsad va vazifalari. Bitiruv malakaviy ishning asosiy maqsadi ba'zi ekstremal masalalarni yechishda graflar nazariyasining tatbiqlari sifatida Deyskra algoritmidan foydalanib, minimal masofaga ega bo'lgan yo'lni aniqlash hamda qiziqarli boshqotirma (kalkulyator) masalasi uchun ob'ektga yo'naltirilgan "Delphi" dasturlash tilida viziullashtirilgan dasturiy vositalarini ishlab chiqishdan iborat.

Ilmiy-tatqiqot usullari. Ushbu bitiruv malakaviy ishida qaralgan masalalarni yechishda grafning minimal uzunligini aniqlash (Deyskra algoritmi),

boshqotirma masalasini yechishda grafning eni bo'yicha qidirish usullaridan foydalanildi.

Mavzuning o'rganilish darajasi. 1736 yilda L. Eyler tomonidan o'sha davrda qiziqarli amaliy masalalardan biri hisoblangan Kyonigsberg ko'priklari haqidagi masalaning qo'yilishi va yechilishi **graflar nazariyasining** paydo bo'lishiga asos bo'ldi.

XIX asrning o'rtalarida graflar nazariyasi bilan bog'liq tadqiqotlar G. Kirxgof va A. Keli ishlarida paydo bo'ldi.

"Graf" iborasi D. Kyonig tomonidan 1936 yilda graflar nazariyasiga bag'ishlangan dastlabki darslikda uchraydi.

Graflar nazariyasi bo'yicha tadqiqotlar natijalari inson faoliyatining turli sohalarida qo'llaniladi. Ulardan ba'zilar quyidagilardir: boshqotirmalarni hal qilish; qiziqarli o'yinlar; yo'llar, elektr zanjirlari, integral sxemalari va boshqarish sistemalarini loyihalashtirish; avtomatlar, blok-sxemalar va komp'yuter uchun programmalarni tadqiq qilish va hokazo.

Tadqiqotning ilmiy yangiligi. Bitiruv malakaviy ishida olingan natijalar ilmiy-uslubiy xarakterga ega bo'lib, ba'zi ekstremal masalalarni yechishda graflar nazariyasining tatbiqlari sifatida Deysktra algoritmidan foydalanib, minimal masofaga ega bo'lgan yo'lni aniqlash hamda qiziqarli boshqotirma (kalkulyator) masalasi uchun dasturiy vositalari ishlab chiqilgan.

Tadqiqot predmeti va ob'ekti. Tadqiqotning predmeti graflar nazariyasi bo'lib, ob'ekti esa minimal masofaga ega bo'lgan yo'lni aniqlash hamda qiziqarli boshqotirma (kalkulyator) masalasidan iborat.

Tadqiqotning ilmiy va amaliy ahamiyati. Ishda olingan natijalar va unda qo'llanilgan usullardan turli iqtisodiy, ijtimoiy sohalar, graflar nazariyasi va turli boshqotirma masalalarni hamda "Informatika" fani bo'yicha o'tkaziladigan fan olimpiadalarining shu turdagi masalalarini yechishda foydalanish mumkin.

Ishning tuzilishi. Ushbu ish kirish, ikki bob, xulosa, foydalanilgan adabiyotlar ro'yxati va ilovalardan iborat.

I bob 6 ta paragrafdan iborat bo'lib, unda olingan natijalarni bayon qilishda zarur bo'lgan asosiy tushunchalar: graflar haqida qisqa tarixiy ma'lumotlar, grafning abstrakt matematik tushuncha sifatidagi ta'rifi va u bilan bog'liq boshlang'ich tushunchalar, graflarning geometrik ravishda, qo'shnilik va insidentlik matritsalarini vositasida berilishi, grafning elementlari ustida sodda amallar, graflarni birlashtirish, biriktirish va ko'paytirish amallari, marshrutlar va zanjirlar, grafning bog'lamliligi tushunchasi, Eyer va Gamilton graflari haqida ma'lumotlar berilgan.

II bobda ishga oid graflar nazariyasining bir necha masalalari minimal masofaga ega bo'lgan yo'lni aniqlash hamda qiziqarli boshqotirma (kalkulyator) masalasi qaralib, ular Deysktra algoritmi, boshqotirma masalasini yechishda grafning eni bo'yicha qidirish algoritmidan foydalanib, ular uchun ob'ektga yo'naltirilgan "Delphi" dasturlash tilida, viziullashtirilgan dasturiy vositalari ishlab chiqilgan va dasturdan foydalanuvchilar uchun ko'rsatmalar berilgan.

Olingan natijalarning qisqacha mazmuni. Bitiruv malakaviy ishida qaralgan masalalar va ulardan olingan asosiy natijalarni qisqacha keltirib o'tamiz:

2.1-masala. Boshlang'ich va oxirgi uchlari hamda yoylari uzunligi bilan berilgan uchlari soni N ta bo'lgan graflarda eng qisqa masofaga ega bo'lgan marshrut hamda uning yo'li uzunligini aniqlovchi dastur tuzilsin.

2.2-masala. Uchlari soni N ta bo'lgan graflarda yoylari uzunligi, boshlang'ich va oxirgi uchlarni kiritish orqali berilgan grafdagi eng qisqa yo'l (marshrut) uzunligini aniqlovchi dastur tuzilsin.

2.3-masala (Kalkulyator). Quyidagi amallarni bajaruvchi kalkulyator berilgan:

X sonini 2 ga ko'paytirish;

X sonini 3 ga ko'paytirish;

X soniga birni qo'shish;

1 sonidan N sonini hosil qilish uchun kamida nechta amal bajarish kerak.

Kiruvchi ma'lumotlar:

1000000 dan oshmaydigan N soni berilgan.

Chiquvchi ma'lumotlar:

Birinchi satrda minimal amallar sonini chiqaring.

Bu masalalarni yechish uchun ob'ektga yo'naltirilgan "Delphi" dasturlash tilida qulay interfeysga ega bo'lgan viziullashtirilgan dasturiy vositalar ishlab chiqildi (1-3-ilovalarga qarang).

Bu dasturiy vositalardan foydalanish natijasida xususiy hollar uchun natijalar quyidagilardan iborat.

2.1-masalada boshlanishi 1-uch oxiri 5-uch yoylari uzunliklari esa 1-uchdan 2- uchgacha bo'lgan masofa 15, 2-uchdan 3- uchgacha bo'lgan masofa 10, 2-uchdan 4- uchgacha bo'lgan masofa 8, 3-uchdan 1- uchgacha bo'lgan masofa 12, 3-uchdan 5- uchgacha bo'lgan masofa 4, 4-uchdan 5- uchgacha bo'lgan masofa 8 ga teng bo'lgan grafda eng qisqa masofaga ega bo'lgan marshrut 1-3-5 hamda uning yo'li uzunligi 19 ga teng bo'lishi aniqlandi (2.3-chizmaga qarang).

2.3-masalada 25 soni uchun masala shartiga ko'ra 5 ta amal bajarilar;

479 soni uchun masala shartiga ko'ra 11 ta amal bajarilar ekan.

I BOB. Graflar nazariyasining asosiy tushunchalari

Ushbu bobda graflar haqida qisqa tarixiy ma'lumotlar, grafning abstrakt matematik tushuncha sifatidagi ta'rifi va u bilan bog'liq boshlang'ich tushunchalar, graflarning geometrik ravishda, qo'shnilik va insidentlik matritsalarini vositasida berilishi, grafning elementlari ustida sodda amallar, graflarni birlashtirish, birlashtirish va ko'paytirish amallari, marshrutlar va zanjirlar, grafning bog'lamliligi tushunchasi, Eyler va Gamilton graflari haqida ma'lumotlar berilgan.

1.1. Graflar nazariyasining boshlang'ich ma'lumotlari

Graflar nazariyasi haqida umumiy ma'lumotlar. 1736 yilda L. Eyler tomonidan o'sha davrda qiziqarli amaliy masalalardan biri hisoblangan Kyonigsberg ko'priklari haqidagi masalaning qo'yilishi va yechilishi **graflar nazariyasining** paydo bo'lishiga asos bo'ldi.

Grafning abstrakt ta'rifi va u bilan bog'liq boshlang'ich tushunchalar. Avvalo, grafning abstrakt matematik tushuncha sifatidagi ta'rifini va boshqa ba'zi sodda tushunchalarni keltiramiz. V qandaydir bo'shmas to'plam bo'lsin. Uning $v_1 \in V$ va $v_2 \in V$ elementlaridan tuzilgan $\langle v_1, v_2 \rangle$ ko'rinishdagi barcha juftliklar (kortejlar) to'plamini (V to'plamning o'z-o'ziga Dekart ko'paytmasini) $V \times V$ bilan belgilaymiz.

Graf deb shunday $\langle V, U \rangle$ juftlikka aytiladiki, bu yerda $V \neq \emptyset$ va $U - \langle v_1, v_2 \rangle (v_1 \in V, v_2 \in V)$ ko'rinishdagi juftliklar korteji bo'lib, $V \times V$ to'plamning elementlaridan tuzilgandir.

Bundan buyon grafni belgilashda $\langle V, U \rangle$ yozuv o'rniga (V, U) yozuvdan foydalanamiz. Grafning tashkil etuvchilarini ko'rsatish muhim bo'lmasa, u holda uni lotin alifbosining bitta harfi, masalan, G bilan belgilaymiz.

$G = (V, U)$ graf berilgan bo'lsin. V to'plamning elementlariga G **grafning uchlari**, V to'plamning o'ziga esa, **graf uchlari to'plami** deyiladi.

Graflar nazariyasida “uch” iborasi o‘rniga, ba’zan, **tugun** yoki **nuqta** iborasi ham qo‘llaniladi. Umuman olganda, hanuzgacha graflar nazariyasining ba’zi iboralari bo‘yicha umumiy kelishuv qaror topmagan. Shuning uchun, bundan keyingi ta’riflarda, imkoniyat boricha, muqobil (alternativ) iboralarni ham keltirishga harakat qilamiz.

$G=(V,U)$ grafning ta’rifiga ko‘ra, U bo‘sh kortej bo‘lishi ham mumkin. Agar U bo‘sh bo‘lmasa, u holda bu kortej (a,b) ($a \in V, b \in V$) ko‘rinishdagi juftliklardan tashkil topadi, bunda $a=b$ bo‘lishi hamda ixtiyoriy (a,b) juftlik U kortejda istalgancha marta qatnashishi mumkin.

$(a,b) \in U$ juftlikni tashkil etuvchi a va b uchlarning joylashish tartibidan bog‘liq holda, ya’ni yo‘nalishning borligi yoki yo‘qligiga qarab, uni turlicha atash mumkin. Agar (a,b) juftlik uchun uni tashkil etuvchilarning joylashish tartibi ahamiyatsiz, ya’ni $(a,b)=(b,a)$ bo‘lsa, (a,b) juftlikka **yo‘naltirilmagan (oriyentirlanmagan) qirra** (yoki, qisqacha, **qirra**) deyiladi. Agar bu tartib muhim, ya’ni $(a,b) \neq (b,a)$ bo‘lsa, u holda (a,b) juftlikka **yoy** yoki **yo‘naltirilgan (oriyentirlangan) qirra** deyiladi.

U kortejning tarkibiga qarab, uni yo **grafning qirralari korteji**, yo **yoylari korteji**, yoki **qirralari va yoylari korteji** deb ataymiz.

Grafning uchlari va qirralari (yoylari) uning **elementlari** deb ataladi. $G=(V,U)$ graf elementlarining soni $(|V|+|U|)$ ga tengdir, bu yerda G grafning uchlari soni $|V| \neq 0$ va $|U|$ bilan uning qirralari (yoylari) soni belgilangan.

Grafning qirrasi (yoyi), odatda, uni tashkil etuvchi uchlar yordamida (a,b) , yoki ab , yoki $(a;b)$ ko‘rinishda belgilanadi. Boshqa belgilashlar ham ishlatiladi: masalan, yoy uchun $\overrightarrow{(a,b)}$ yoki $\overleftarrow{(a,b)}$, qirra uchun $\overleftrightarrow{(a,b)}$, yoy yoki qirra uchun u (ya’ni uchlari ko‘rsatilmasdan bitta harf vositasida) ko‘rinishda.

Graf yoyi uchun uning chetki uchlarini ko‘rsatish tartibi muhim ekanligini ta’kidlaymiz, ya’ni (a,b) va (b,a) yozuvlar bir-biridan farq qiluvchi yoylarni ifodalaydi. Agar yoy (a,b) ko‘rinishda ifodalangan bo‘lsa, u holda a uning **boshlang‘ich uchi**, b esa **oxirgi uchi** deb ataladi. Bundan tashqari, yoy (a,b)

ko‘rinishda yozilsa, u haqida a **uchdan chiquvchi (boshlanuvchi)** va b **uchga kiruvchi (uchda tugovchi)** yoy deb aytish ham odat tusiga kirgan.

Qirra uchun uning (a,b) yozuvidagi harflar joylashish tartibi muhim rol o‘ynamaydi va a va b elementlar **qirraning uchlari** yoki **chectlari** deb ataladi.

Agar grafda yo (a,b) qirra, yo (a,b) yoy, yoki (b,a) yoy topilsa, u holda a va b **uchlar tutashtirilgan** deyiladi. Agar grafning ikkita uchini tutashtiruvchi qirra yoki yoy bor bo‘lsa, u holda ular **qo‘shni uchlar** deb, aks holda esa, **qo‘shni bo‘lmagan uchlar** deb aytiladi.

Grafning ikkita uchi qo‘shni bo‘lsa, ular shu uchlarni tutashtiruvchi **qirraga (yoyga) insident**, o‘z navbatida, qirra yoki yoy bu **uchlarga insident** deyiladi.

Grafda ikkita qirra (yoy) umumiy chetga ega bo‘lsa, ular **qo‘shni qirralar (yoylar)** deyiladi.

Shuni ta’kidlash kerakki, qo‘shnilik tushunchasi grafning bir jinsli, insidentlik tushunchasi esa uning turli jinsli elementlari orasidagi munosabatni ifodalaydi.

Ba’zan graf undagi elementlar soniga qarab, ya’ni **uchlar soni** m va **qirralar (yoylar) soni** n ga qarab belgilanadi va bu holda grafni (m,n) -**graf** deb ataydilar.

Agar $G=(V,U)$ grafda U kortej faqat qirrallardan iborat bo‘lsa, u holda **yo‘naltirilmagan (oriyentirlanmagan)** va faqat yo‘naltirilgan (oriyentirlangan) qirrallardan (ya’ni, yoylardan) tashkil topgan bo‘lsa, u holda u **yo‘naltirilgan (oriyentirlangan) graf** deb ataladi. Oriyentirlangan graf, qisqacha, **orgraf** deb ham ataladi.

Ko‘p hollarda oriyentirlanmagan qirrallari ham, oriyentirlangan qirrallari ham bo‘lgan graflar bilan ish ko‘rishga to‘g‘ri keladi. Bunday graflar **aralash graflar** deb ataladi.

Agar $G=(V,U)$ grafning (orgrafning) U korteji tarkibida $V \times V$ to‘plamdan olingan takrorlanuvchi elementlar bo‘lsa, u holda ular **karrali** yoki **parallel**

qirralar (yoylar) deb ataladi. Karrali qirralari yoki yoylari bo‘lgan graf **multigraf** deyiladi.

Ikkala chetki (boshlang‘ich va oxirgi) uchlari ustma-ust tushgan qirra (yoy), ya‘ni grafning $(a,a) \in U$ elementi **sirtmoq** deb ataladi. Sirtmoq, odatda, yo‘naltirilmagan deb hisoblanadi. Qirralari (yoylari) orasida sirtmoqlari bo‘lgan graf **psevdograf** deyiladi.

Umumiy holda uchlar to‘plami V va (yoki) qirralar (yoylar, qirra va yoylar) kortaji U cheksiz ko‘p elementli bo‘lishi mumkin. Bundan keyin V to‘plam va U kortej faqat chekli bo‘lgan $G=(V,U)$ graflarni qaraymiz. Bunday graflar **chekli graflar** deb ataladi.

Hech qanaqa qirra (yoy) bilan bog‘lanmagan uch **yakkalangan (ajralgan, xolis, yalong‘och) uch** deb ataladi.

Faqat yakkalangan uchlardan tashkil topgan graf (ya‘ni, grafda qirralar va yoylar bo‘lmasa) **nolgraf** yoki **bo‘sh graf** deb ataladi. Uchlari soni m ga teng bo‘lgan bo‘sh grafni O_m yoki N_m kabi belgilash qabul qilingan.

Istalgan ikkita uchlari qo‘shni bo‘lgan sirtmoqsiz va karrali qirralarsiz oriyentirlanmagan graf **to‘la graf** deb ataladi. Uchlari soni m ga teng bo‘lgan to‘la graf K_m bilan belgilanadi. Ravshanki, K_m grafning qirralar soni $C_m^2 = \frac{m(m-1)}{2}$ bo‘ladi.

Agar orgrafning istalgan ikkita uchini har bir yo‘nalishda tutashtiruvchi faqat bittadan yoy mavjud bo‘lsa, u holda unga **to‘la orgraf** deb ataladi. Ravshanki, to‘la grafdagi qirralarning har birini ikkita (yo‘nalishlari bir-biriga qarama-qarshi bo‘lgan) yoylarga almashtirilsa, natijada to‘la orgraf hosil bo‘ladi. Shuning uchun, to‘la orgrafdagi yoylar soni oriyentirlanmagan to‘la grafdagi qirralar sonidan ikki baravar ko‘pdir, ya‘ni uchlari m ta bo‘lgan to‘la orgrafdagi yoylar soni $2C_m^2 = m(m-1)$ bo‘ladi.

Agar grafning uchlariga qandaydir belgilar, masalan, $1,2,...,m$ sonlari mos qo‘yilgan bo‘lsa, u **belgilangan graf** deb ataladi.

Agar $G=(V,U)$ va $G'=(V',U')$ graflarning uchlari to'plamlari, ya'ni V va V' to'plamlar orasida uchlarning qo'shnilik munosabatini saqlaydigan o'zaro bir qiymatli moslik o'rnatish mumkin bo'lsa, u holda G va G' graflar **izomorf graflar** deb ataladi. Bu ta'rifni quyidagicha ham ifodalash mumkin: agar $\forall x,y \in V$ va $x',y' \in V'$ ($x \leftrightarrow y, x' \leftrightarrow y'$) uchun $xy \leftrightarrow x'y'$ ($xy \in U, x'y' \in U'$) bo'lsa, u holda G va G' graflar izomorfdir. Agar izomorf graflardan biri oriyentirlangan bo'lsa, u holda ikkinchisi ham, albatta, oriyentirlangan bo'lishi va ulardagi mos yoylarning yo'nalishlari ham bir-birlariga mos bo'lishlari shart.

Graf uchiga insident qirralar soni shu **uchning lokal darajasi**, yoki, qisqacha, **darajasi**, yoki **valentligi** deb ataladi. Grafdagi a uchning darajasini $\rho(a)$ bilan belgilaymiz.

Sirtmoqqa insident bo'lgan uchning darajasini aniqlashda shuni e'tiborga olish kerakki, qaralayotgan masalaga bog'liq holda sirtmoqni bitta qirra deb ham, ikkita qirra deb ham hisoblash mumkin. Ravshanki, ajralgan uchning darajasi nolga teng. Darajasi birga teng uch **chetki** (yoki **osilgan**) **uch** deb ataladi. Chetki (osilgan) uchga insident qirra ham **chetki** (yoki **osilgan**) **qirra** deb ataladi.

Agar grafning barcha uchlari bir xil r darajaga ega bo'lsa, u holda bunday graf r **darajali regulyar graf** deb ataladi. Uch darajali regulyar graf **kubik** (yoki **uch valentli**) **graf** deb ataladi. O_m graf nol darajali regulyar graf ekanligini, K_m esa $(m-1)$ darajali regulyar graf ekanligini ta'kidlaymiz.

Ko'rinib turibdiki, oriyentirlanmagan grafda barcha uchlar darajalarining yig'indisi qirralar sonining ikki baravariga teng juft son bo'ladi, chunki qirralarni sanaganda har bir qirra hisobda ikki marta qatnashadi. Shunday qilib, XVIII asrdayoq L. Eyler tomonidan isbotlangan quyidagi tasdiq o'rinlidir.

1.1-lemma ("ko'rishishlar" haqida). *Ixtiyoriy oriyentirlanmagan grafda barcha uchlar darajalari yig'indisi qirralar sonining ikki baravariga teng.*

Agar grafning uchlar to'plamini o'zaro kesishmaydigan shunday ikkita qism to'plamlarga (bo'laklarga) ajratish mumkin bo'lsaki, grafning ixtiyoriy qirrasi bu to'plamlarning biridan olingan qandaydir uchni ikkinchi to'plamdan olingan biror

uch bilan tutashtiradigan bo'lsa, u holda bunday graf **ikki bo'lakli graf** (**bixromatik** yoki **Kyonig grafi**) deb ataladi. Ta'rifdan ko'rinib turibdiki, ikki bo'lakli grafning har bir bo'lagidagi ixtiyoriy ikkita uchlar qo'shni bo'la olmaydi. Biror bo'lagida faqat bitta uch bo'lgan to'la ikki bo'lakli graf **yulduz** deb ataladi.

Agar ikki bo'lakli grafning turli bo'laklariga tegishli istalgan ikkita uchi qo'shni bo'lsa, u holda bu graf **to'la ikki bo'lakli graf** deb ataladi. To'la ikki bo'lakli grafni $K_{m,n}$ bilan belgilaymiz, bu yerda m va n bilan grafning bo'laklaridagi uchlar sonlari belgilangan. $K_{m,n} = (V, U)$ graf uchun $|V| = m + n$ va $|U| = mn$ bo'lishi ravshan, bu yerda $|V|$ – $K_{m,n}$ grafning uchlari soni, $|U|$ – uning qirralari soni.

Grafning ikki bo'lakli graf bo'lishi haqidagi ba'zi qo'shimcha ma'lumotlar (Kyonig teoremasi) ushbu bobning 4- paragrafida keltirilgan.

Ikkidan katta ixtiyoriy natural k son uchun k **bo'lakli graf** tushunchasini ham kiritish mumkin.

1.2. Graflarning berilish usullari

Grafning geometrik ifodalanishi. Graflarning turlicha berilish usullari mavjud. Grafning abstrakt matematik ta'rifi uning berilish usullaridan biridir. Grafning abstrakt matematik ta'rifi uni tasavvur qilish, anglash, uning xossalarini o'rganish va bu xossalarni amalda qo'llash jarayonida ba'zi qiyinchiliklar tug'dirishi tabiiydir. Shuning uchun grafning boshqa berilish usullaridan ham foydalaniladi. Masalan, grafning elementlarini, ya'ni uchlari va qirralarini (yoylarini) yozish yoki aytish grafning berilish usuli sifatida qaralishi mumkin. Albatta, grafning yana boshqa berilish usullari ham mavjud. Quyida bu usullarning bir nechasi bilan tanishamiz.

Grafning uchlari tekislikda yoki fazoda nuqtalar bilan, qirralarini (yoylarini) esa mos uchlarni tutashtiruvchi uzluksiz chiziqlar bilan ifodalab, qandaydir diagrammaga – **grafning ko'rgazmali tasviriga** ega bo'lamiz. Agar

uchlar to'plami va bu uchlarning tutashishlarini ko'rgazmali qilib taqdim qilish kerak bo'lsa, grafning geometrik tasvirlanishiga mos shaklni qog'ozda chizib grafni tasvirlash mumkin.

Shuni ta'kidlaymizki, ba'zi hollarda diagrammada graf uchlari doirachalar yordamida yoki qandaydir boshqa usulda ifodalanadi. Grafning qirralariga (yoylariga) mos chiziqlarning to'g'ri yoki egri bo'lishi va ularning uzunligi ahamiyatga ega emas. Muhimi, bu chiziqlar uzluksiz bo'lib, grafning qandaydir ikkita uchlarni tutashtirishi lozim. Agar qirra yo'nalishga ega bo'lsa (ya'ni u yoy bo'lsa), u holda bunday qirrani ifodalovchi chiziqda yo'nalish biror usul bilan, masalan, strelka bilan ko'rsatiladi.

Ixtiyoriy graf uchun bunday diagrammalarni istalgancha tuzish mumkinligi ravshan. Agar biror diagrammada grafning uchlariga mos keluvchi nuqtalar ustma-ust tushmasa, qirralarga mos keluvchi chiziqlar, chetki nuqtalarni hisobga olmaganida, umumiy nuqtalarga ega bo'lmasa, bunday diagramma **grafning geometrik ifodalanishi** deyiladi. Shuni ta'kidlash kerakki, bitta graf turlicha geometrik ifodalanishi mumkin.

Graflar izomorfligining ta'rifi va grafni geometrik ifodalashning mohiyatidan kelib chiqadiki, abstrakt ta'rif yordamida ifodalangan graf va uning geometrik ifodalanishi o'zaro izomorf bo'ladi. Tabiiyki, izomorf graflar turlicha geometrik ifodalanishlari mumkin.

1.1-teorema. *Har qanday chekli grafni 3 o'lchovli Evklid fazosida geometrik ifodalash mumkin.*

Shuni ham ta'kidlash kerakki, 1- teoremadagi 3ni 2ga almashtirib bo'lmaydi, chunki tekislikda qirralarini (yoylarini) ifodalovchi kesishmaydigan (aniqrog'i, chetki nuqtalaridan boshqa umumiy nuqtalari bo'lmagan) chiziqlar yordamida tasvirlash imkoniyati faqat ba'zi graflargagina xos, ya'ni har qanday grafning 2 o'lchovli Evklid fazosida (tekislikda) geometrik ifodalanishi mavjud bo'lavermaydi.

Agar graf tekislikda geometrik ifodalanishga ega bo'lsa, u holda bunday graf **tekis (yassi) graf** deb ataladi. Bunday graf **tekislikda yotuvchi graf** deb ham atalishi mumkin.

Boshqacha aytganda, tekis grafning barcha uchlari bir tekislikda yotadi hamda barcha qirralari (yoylari) o'sha tekislikda yotuvchi o'zaro kesishmaydigan uzluksiz chiziqlar bo'lib, ular faqat o'zlari insident bo'lgan uchlardagina umumiy nuqtalarga ega.

Tekis grafga izomorf graf **planar graf** deb ataladi.

Grafning maxsus turdagi ko'phad yordamida berilishi. Grafni maxsus turdagi ko'phad yordamida ham berish mumkinligini ta'kidlaymiz. Uchlari to'plami $V = \{v_1, v_2, \dots, v_m\}$ bo'lgan G graf berilgan bo'lsin. G grafning yakkalangan uchlari yo'q deb faraz qilamiz,. Bu grafni m ta $x_1, x_2, \dots, x_m$ o'zgaruvchilarga bog'liq

$$f(G) = x_1^{\sigma_1} x_2^{\sigma_2} \dots x_m^{\sigma_m} \prod_{i < j} (x_j - x_i)^{\alpha_{ij}}$$

ko'rinishdagi ko'phad yordamida tasvirlash mumkin, bu yerda ko'paytma $i < j$ shartni qanoatlantiruvchi barcha (i, j) juftlar bo'yicha amalga oshiriladi, x_i o'zgaruvchi $v_i \in V$ uchga mos keladi, α_{ij} – v_i va v_j uchlarni tutashtiruvchi qirralar soni, σ_i – v_i uchdagi sirtmoqlar soni.

Qo'shnilik matritsalar. Endi grafning boshqa bir berilish usuli negizida yotuvchi **graf uchlari qo'shniligi matritsasi** tushunchasini qarab chiqamiz.

$G = (V, U)$ – uchlari soni m ga teng bo'lgan belgilangan, sirtmoqsiz va karrali qirralarsiz graf bo'lsin.

Elementlari
$$a_{ij} = \begin{cases} 1, & \text{agar } i \text{ va } j \text{ uchlar qo'shni bo'lsa,} \\ 0, & \text{aks holda.} \end{cases}$$

ko'rinishda aniqlangan $A = (a_{ij})$ ($i = 1, 2, \dots, m$; $j = 1, 2, \dots, m$) matritsani grafning uchlari qo'shniligi matritsasi deb ataymiz.

Bu ta'rifdan sirtmoqsiz va karrali qirralari bo'lmagan graf uchlari qo'shniligi matritsasining bosh diagonalida faqat nollar bo'lishi, satrlaridagi birlar soni esa mos uchlarning darajalariga tengligi kelib chiqadi.

Uchlari soni m ga teng bo'lgan belgilangan **oriyentirlangan** $G=(V,U)$ **grafning uchlari qo'shniligi** $m \times m$ -**matritsasi** deb elementlari

$$a_{ij} = \begin{cases} 1, & \text{agar } (i, j) \in U \text{ bo'lsa,} \\ 0, & \text{aks holda,} \end{cases}$$

ko'rinishda aniqlangan $A=(a_{ij})$ ($i=1,2,\dots,m$, $j=1,2,\dots,m$) matritsaga aytiladi.

Endi G uchlari $1,2,\dots,m$ bo'lgan belgilangan oriyehtirlanmagan multigraf bo'lsin. a_{ij} elementlari G grafning i va j uchlarini tutashtiruvchi qirralar soniga teng bo'lgan $A=(a_{ij})$ ($i, j=1,2,\dots,m$) matritsa **oriyentirlanmagan multigrafning uchlari qo'shniligi matritsasi** deb ataladi.

Karrali yoylari bo'lgan **sirtmoqsiz orgraf uchlari qo'shniligi matritsasi** tushunchasini ham yuqoridagiga o'xshash ta'riflash mumkin.

1.2- teorema. *Graflar faqat va faqat uchlari qo'shniligi matritsalarini bir-birlaridan satrlarining o'rinlarini va ustunlarining o'rinlarini mos almashtirishlar yordamida hosil bo'lsagina izomorf bo'lishadi.*

Shunday qilib, manfiymas butun sonlardan tashkil topgan va graf uchun uchlari qo'shniligi matritsasi bo'lgan kvadrat matritsa bilan graf orasida bir qiymatli moslik (izomorflik aniqligida) bor degan xulosa va, bundan, graflar nazariyasi bo'yicha izlanishlar maxsus shartlarni qanoatlantiruvchi matritsalarini tadqiq qilishga keltirilishi mumkinligi kelib chiqadi.

$u_1, u_2, \dots, u_n$ ($n \geq 1$) qirralarga ega yakkalangan uchlari, sirtmoq va karrali qirralari bo'lmagan graf uchun elementlari

$$c_{ij} = \begin{cases} 1, & \text{agar } u_i \text{ va } u_j \text{ qirralar umumiy uchga ega bo'lsa,} \\ 0, & \text{agar } u_i = u_j \text{ bo'lsa yoki ularning umumiy uchi bo'lmasa,} \end{cases}$$

quyidagicha aniqlangan $C=(c_{ij})$ ($i=1,2,\dots,n$, $j=1,2,\dots,n$) $n \times n$ -matritsa **grafning qirralari qo'shniligi matritsasi** deb ataladi.

Ravshanki, sirtmoqsiz va karrali qirralarsiz graf qirralari qo'shniligi matritsasi bosh diagonalga nisbatan simmetrik kvadrat matritsadir va uning bosh diagonali nollardan iborat.

Insidentlik matritsalar. Uchlari $1, 2, \dots, m$ va qirralari $u_1, u_2, \dots, u_n$ ($n \geq 1$) bo'lgan belgilangan graf berilgan bo'lsin. Bu grafning uchlariga satrlari, qirralariga esa ustunlari mos keluvchi va elementlari

$$b_{ij} = \begin{cases} 1, & \text{agar } i \text{ uch } u_j \text{ qirraga insident bo'lsa,} \\ 0, & \text{agar } i \text{ uch } u_j \text{ qirraga insident bo'lmasa,} \end{cases}$$

ko'rinishda aniqlangan $B = (b_{ij})$ ($i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$) matritsa grafning **insidentlik matritsasi** deb ataladi.

Endi uchlari $1, 2, \dots, m$ va qirralari $u_1, u_2, \dots, u_n$ ($n \geq 1$) bo'lgan belgilangan sirtmoqsiz orgrafni qaraymiz. Elementlari

$$b_{ij} = \begin{cases} 1, & \text{agar } i \text{ uch } u_j \text{ yoyning boshlang'ich uchi bo'lsa,} \\ -1, & \text{agar } i \text{ uch } u_j \text{ yoyning oxirgi uchi bo'lsa,} \\ 0, & \text{agar } i \text{ uch va } u_j \text{ yoy insident bo'lmasa.} \end{cases}$$

ko'rinishda aniqlangan $B = (b_{ij})$ ($i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$) matritsaga grafning **insidentlik matritsasi** deb ataladi.

1.3- teorema. *Graflar (orgraflar) faqat va faqat insidentlik matritsalarini bir-birlaridan satrlarining o'rinlarini va ustunlarining o'rinlarini mos almashtirishlar yordamida hosil bo'lsagina izomorf bo'lishadi.*

1.3. Graflar ustida amallar

Graflar ustida sodda amallar. Graflar ustida turli amallar bajarish mumkin, masalan, graflarni birlashtirish, birlashtirish, ko'paytirish, grafni qismlarga ajratish va hokazo.

Eng sodda amallardan biri sifatida grafdan **uchni olib tashlash** amalini keltirsa bo'ladi. Bu amalni qo'llash berilgan grafning uchlari to'plamidan birorta element yo'qotishni (olib tashlashni) anglatadi. Natijada uchlari soni bittaga kamaygan yangi graf hosil bo'ladi. Albatta, bu amalni uchlari soni ikkitadan kam bo'lmagan graflar uchun qo'llash mumkin bo'lib, uni bajarish jarayonida olib

tashlanayotgan uch bilan birgalikda shu uchga insident bo'lgan barcha qirralar (yoylar) ham olib tashlanadi.

Eng sodda amallar qatoriga grafdan **qirrani (yoyni) olib tashlash** amalini ham kiritish mumkin. Bu amalga ko'ra berilgan grafning qirralari (yoylari) to'plamidan birorta element yo'qotiladi (olib tashlanadi). Berilgan grafdan qirrani (yoyni) olib tashlayotganda shu qirraga (yoyga) insident uchlarni grafda qoldirish ham yo'qotish ham mumkin. Bu yerda vaziyatga qarab ish yuritiladi.

$G = (V, U)$ va $G' = (V', U')$ graflar berilgan bo'lsin. Agar $V \subseteq V'$ va G grafning barcha qirralari (yoylari) G' grafning ham qirralari (yoylari), ya'ni $U \subseteq U'$ bo'lsa, u holda G graf G' grafning **qism grafi** deb ataladi.

Agar G graf karrali qirralarga ega bo'lmasa, u holda uchlari G grafning barcha uchlari iborat bo'lgan shunday yagona $\overline{G}$ graf mavjudki, $\overline{G}$ grafdagi barcha juft uchlari faqat va faqat G grafda qo'shni bo'lmagandagina qo'shnidir. Bunday $\overline{G}$ graf berilgan G grafning **to'ldiruvchi grafi** deb ataladi.

Berilgan graf uchun to'ldiruvchi grafni qurish jarayonini ham graflar ustida bajariladigan amallar qatoriga kiritish mumkin. G graf uchun **to'ldiruvchi grafni qurish** amalini qo'llash natijasida $\overline{G}$ graf hosil bo'ladi. Isbotlash mumkinki, $\overline{\overline{G}} = G$ munosabat o'rinlidir.

Graflar ustida shunday amallarni bajarish mumkinki, ular elementlari soni berilgan grafdagidan ko'proq bo'lgan boshqa graflarning hosil bo'lishiga olib keladi. Bunday amallar qatoriga **uchni qo'shish amali** yoki **qirrani (yoyni) qo'shish amalini** kiritish mumkin.

Grafga yangi uchni qo'shish turlicha usul bilan amalga oshirilishi mumkin. Masalan, yangi v uchni berilgan grafga qo'shish shu grafning v_1 va v_2 uchlari insident bo'lgan qandaydir u qirrasiga qo'shish orqali quyidagicha ikki bosqichda bajarilishi mumkin:

- 1) u qirra berilgan grafdan olib tashlanadi;
- 2) hosil bo'lgan grafga ikkita yangi qirralar: v va v_1 uchlarga insident u_1 qirra hamda v va v_2 uchlarga insident u_2 qirra qo'shiladi.

Bu jarayon grafda **qirraga darajasi 2 bo'lgan yangi uchni qo'shish (kiritish)** yoki **qirrani ikkiga bo'lish amali** deb ataladi.

Agar G graf G' grafdan qirrani ikkiga bo'lish amalini chekli marta ketma-ket qo'llash vositasida hosil qilingan bo'lsa, u holda G **graf G' grafning bo'linish grafi** deb ataladi.

Bo'linish graflari izomorf bo'lgan graflar **gomeomorf graflar** deb ataladi.

Graflarni birlashtirish. $G_1 = (V_1, U_1)$ va $G_2 = (V_2, U_2)$ graflar berilgan bo'lsin. Uchlari to'plami $V = V_1 \cup V_2$ va qirralari (yoylari) korteji $U = U_1 \cup U_2$ kabi aniqlangan $G = (V, U)$ graf G_1 va G_2 **graflarning birlashmasi (uyushmasi)** deb ataladi va $G = G_1 \cup G_2$ ko'rinishda belgilanadi.

Agar birlashtirilayotgan graflarning uchlari to'plamlari kesishmasa, u holda bu graflarning birlashmasi **diz'yunkt birlashma** deb ataladi.

Graflarni biriktirish. $G_1 = (V_1, U_1)$ va $G_2 = (V_2, U_2)$ graflar berilgan bo'lsin. G_1 va G_2 graflar birlashtirilishi hamda G_1 grafning har bir uchi G_2 grafning har bir uchi bilan qirra vositasida tutashtirilishi natijasida hosil bo'lgan $G = (V, U)$ graf G_1 va G_2 **graflarning birikmasi (tutashmasi)** deb ataladi va $G = G_1 + G_2$ ko'rinishda belgilanadi.

Agar uchlari to'plamlari kesishmasi bo'sh bo'lmagan graflarni biriktirish zarur bo'lsa, u holda hal qilinayotgan masala xossalarini e'tiborga olib ish ko'rish kerakligini ta'kidlaymiz.

Graflarni ko'paytirish. $G_1 = (V_1, U_1)$ va $G_2 = (V_2, U_2)$ graflar berilgan bo'lsin. Uchlari to'plami $V = V_1 \times V_2$ bo'lgan $G = (V, U)$ grafning qirralari (yoylari) kortejini quyidagicha aniqlaymiz: agar $v_1' = v_1''$ va $(v_2', v_2'') \in U_2$ yoki $v_2' = v_2''$ va $(v_1', v_1'') \in U_1$ bo'lsa, u holda $(v', v'') \in U$ bo'ladi, bu yerda $v_1', v_1'' \in V_1$, $v_2', v_2'' \in V_2$, $v' = (v_1', v_2') \in V$ va $v'' = (v_1'', v_2'') \in V$. Shunday usul bilan hosol qurilgan $G = (V, U)$ graf G_1 va G_2 **graflarning ko'paytmasi** deb ataladi va $G = G_1 \times G_2$ kabi belgilanadi.

Graflarning ko'paytmasi ta'rifiga asosan berilgan $G_1 = (V_1, U_1)$ va $G_2 = (V_2, U_2)$ graflarning ko'paytmasi hisoblangan G grafdagi:

– uchlar (v_1, v_2) yoki (v_2, v_1) ko‘rinishdagi juftliklardan iboratdir, bu yerda $v_1 \in V_1, v_2 \in V_2$;

– $v' = (v_1', v_2') \in V$ va $v'' = (v_1'', v_2'') \in V$ uchlar faqat va faqat shu holda qo‘shni bo‘ladilarki, qachonki bu uchlarni (juftliklarni) tashkil qiluvchi elementlarning biriunga mos element bilan ustma-ust tushgan holda boshqa elementlar o‘z grafida qo‘shni bo‘lishsa, bu yerda $v_1', v_1'' \in V_1, v_2', v_2'' \in V_2$;

– $|V_1| = m_1, |V_2| = m_2, |U_1| = n_1$ va $|U_2| = n_2$ munosabatlardan $|V| = m_1 m_2$ va $|U| = m_1 n_2 + m_2 n_1$ bo‘lishi kelib chiqadi.

1.4. Marshrutlar va zanjirlar

Marshrutlar va zanjirlar haqida umumiy ma’lumotlar. Uchlari to‘plami $V = \{v_1, v_2, \dots, v_m\}$ va qirralar korteji $U = \{u_1, u_2, \dots, u_n\}$ bo‘lgan oriyentirlanmagan $G = (V, U)$ graf berilgan bo‘lsin. Bu G grafdagi uchlar va qirralarning har ikki qo‘shni qirralari umumiy chetki uchga ega

$$(\dots, v_{i_1}, u_{j_1}, v_{i_2}, u_{j_2}, v_{i_3}, \dots)$$

ko‘rinishdagi chekli yoki cheksiz ketma-ketligi **marshrut** deb ataladi. Marshrutni uning uchlari ketma-ketligi $(\dots, v_{i_1}, v_{i_2}, \dots)$ yoki qirralari ketma-ketligi $(\dots, u_{j_1}, u_{j_2}, \dots)$ ko‘rinishda ham belgilash mumkin.

Agar marshrutda qandaydir uchdan oldin uchlar bo‘lmasa, bu uchni marshrutning **boshlang‘ich uchi** deb, shu uchdan keyin marshrutga tegishli uchlar bo‘lmaganda esa, uni marshrutning **oxirgi uchi** deb ataydilar.

Agar marshrutning boshlang‘ich uchi v_p va oxirgi uchi v_q bo‘lsa, u holda uni v_p **uchdan** v_q **uchga yo‘nalgan marshrut** yoki **chetlari** v_p va v_q **bo‘lgan marshrut** deb ataladi.

Marshrutdagi ikkita qoshni qirralarga tegishli uch **ichki uch** yoki **oraliq uch** deb ataladi.

Marshrutda qirralar va uchlar takrorlanishi mumkin bo‘lgani uchun marshrutning ichki uchi, bir vaqtning o‘zida, uning boshlang‘ich va (yoki) oxirgi

uchi bo‘lishi ham mumkin va teskarisi, marshrutning boshlang‘ich va (yoki) oxirgi uchi uning ichki uchi bo‘lishi ham mumkin.

Tabiiyki, marshrut:

– boshlang‘ich uchga ham oxirgi uchga ham ega bo‘lmasligi mumkin (bunday marshrut **ikki tomonlama cheksiz marshrut** deb ataladi);

– boshlangich uchga ega bo‘lib, oxirgi uchga ega bo‘lmasligi mumkin yoki, aksincha, oxirgi uchga ega bo‘lib, boshlangich uchga ega bo‘lmasligi mumkin (**bir tomonlama cheksiz marshrut**);

– yagona qirradan iborat bo‘lishi mumkin (**notrivial marshrut**);

– birorta ham qirraga ega bo‘lmasligi mumkin (**nol marshrut** yoki **trivial marshrut**).

Marshrutning uzunligi deb undagi qirralar soniga aytiladi.

Turli qirralardan tashkil topgan marshrutga **zanjir** deb ataladi. Agar zanjirning chetlaridan tashqari barcha uchlari turlicha bo‘lsa, u holda uni **oddiy zanjir** deb ataydilar.

Berilgan $(v_1, v_2, \dots, v_s)$ zanjir yoki oddiy zanjir uchun $v_1 = v_s$ bo‘lsa, u **yopiq zanjir** deb ataladi. Hech bo‘lmaganda bitta qirraga ega yopiq oddiy zanjir **sikl** deb ataladi.

Sirtmoq yoki bir juft karrali qirralar sikl tashkil etishi ravshandir.

Tushunarliki, grafdagi zanjir grafning qism grafi deb qaralishi mumkin.

Oriyentirlangan graflar uchun ham undagi yoylarning yo‘nalishini (oriyentatsiyasini) inobatga olmasdan oriyentirlanmagan marshrut, zanjir va oddiy zanjir tushunchalarini kiritish mumkin. Lekin, oriyentirlangan graflar uchun oriyentirlangan marshrut tushunchasini kiritish tabiiydir.

Yoylarning oriyentatsiyalari hisobga olingan yoylar va uchlar ketma-ketligi **oriyentirlangan marshrut** deb ataladi.

Oriyentirlangan marshrut uchun zanjir tushunchasiga o‘xshash **yo‘l** (yoki **oriyentirlangan zanjir**) tushunchasini ham kiritish mumkin. Boshlang‘ich va oxirgi uchlari ustma-ust tushadigan oriyentirlangan zanjir **kontur** deb ataladi.

1.4- teorema. *Agar grafdagi har bir uchning lokal darajasi ikkidan kichik bo'lmasa, u holda bu graf siklga ega.*

Grafning bog'lamliligi tushunchasi. Agar oriyentirlanmagan grafda chetlari a va b uchlardan iborat marshrut topilsa, bu a va b uchlar **bog'langan** deb, marshrutning o'zi esa a va b **uchlarni bog'lovchi marshrut** debataladi.

Tabiiyki, agar qandaydir uchlarni bog'lovchi marshrut biror a_i uchdan bir necha marta o'tsa, u holda marshrutning siklik qismini olib tashlab (bunda siklik qismning o'rniga marshrutda faqat a_i uch qoldiriladi) yana o'sha uchlarni bog'lovchi oddiy zanjir ko'rinishdagi marshrutni hosil qilish mumkin. Shuning uchun, marshrut bilan bog'langan uchlar doimo oddiy zanjir bilan ham bo'glangan bo'ladi degan xulosaga kelamiz.

Bir-biri bilan ustma-ust tushmaydigan ixtiyoriy ikkita uchlari bog'langan graf **bog'lamli graf** deb ataladi.

Agar grafdagi ikkita uchni biror oddiy zanjir bilan tutashtirish mumkin bo'lsa, u holda bu ikkita uch **ekvivalent (bog'langan)** deyiladi. Bunday uchlar to'plami grafda **ekvivalentlik munosabati** bilan aniqlangan deb hisoblanadi. Uchlar to'plami bo'yicha ekvivalentlik munosabatini inobatga olgan holda berilgan grafni **bog'lamlilik komponentalari** (qisqacha, **komponentalari**) deb ataluvchi bog'lamli qismlarning birlashmasi deb qarash mumkin. Bu yerda berilgan graf bog'lamlilik komponentalariga bo'laklandi (ajratildi) deb aytish mumkin. Isbotlash mumkinki, har qanday graf o'zining bog'lamlilik komponentalarining diz'yunktiv birlashmasi sifatida ifodalanishi mumkin, bunda grafning bog'lamlilik komponentalariga bo'laklanishi bir qiymatli aniqlanadi.

Keyingi ma'lumotlarni bayon etish uchun **yoq** tushunchasi zarur bo'ladi. Tekislikda geometrik ifodalanuvchi grafni qaraymiz. Bu grafga tegishli bo'lmagan (ya'ni grafning hech qaysi uchi bilan ustma-ust tushmaydigan va uning hech qaysi qirrasida yotmaydigan) biror A nuqtani hech qaysi nuqtasi grafga tegishli bo'lmagan uzluksiz chiziq bilan tutashtirish mumkin bo'lgan barcha nuqtalar to'plami grafning A nuqtani o'zida saqlovchi **yoqi** deb ataladi.

Yoq tushunchasiga berilgan ta'rifga ko'ra yoq grafning geometrik ifodalanishi yordamida tekislikning "qirqib" olinadigan qismidan iboratdir. Tekislikda geometrik ifodalanuvchi ixtiyoriy grafning hech bo'lmaganda bitta yoqi bo'lishi va uning bitta yoqi chegaraga ega emasligi (cheksizligi) o'z-o'zidan ravshandir.

1.5- teorema (Eylar 1752). *Tekis va bog'lamli $G=(V,U)$ graf uchun $m+r=2+n$ tenglik o'rinlidir, bu yerda $m=|V|$, $n=|U|$, r – yoqlar soni.*

1.5-teoremaning tasdig'idagi $m+r=2+n$ tenglik **Eylar formulasi** deb ataladi.

Eylar teoremasidan bir qator natijalar kelib chiqadi. Masalan, bu teoremadan foydalanib uni osonlik bilan bog'lamli bo'lmagan graflar uchun quyidagicha umumlashtirish mumkin.

Tabiiyki, bog'lamli grafdan qirrani yoki bir necha qirralarni olib tashlash natijasida hosil bo'lgan graf bog'lamli bo'lishi ham bo'lmasligi ham mumkin. Agar bog'lamli grafdan qirrani olib tashlash amali grafning bog'lamlilik xususiyatini buzsa, u holda bunday qirrani **ajratuvchi qirra** deb ataymiz.

Ravshanki, berilgan bog'lamli grafda ajratuvchi qirralar ko'p bo'lishlari mumkin. Ajratuvchi qirralar to'plamining hech qaysi qism to'plami elementlari ajratuvchi qirralar bo'lmasa, bu qirralar to'plamini **kesim** deb ataymiz. Grafdan kesimga tegishli qirralar olib tashlansa, natijada ikki bog'lamli komponentalari bo'lgan graf hosil bo'lishi ravshandir. Agar kesim yagona qirradan iborat bo'lsa, u holda bu qirra **ko'pri**k deb ataladi.

Bog'lamli graf uchlarini belgilash jarayoni tugagandan so'ng, uning uchlari to'plami V ni ikkita V_j va V_q to'plamga quyidagicha ajratamiz: juft raqamli uchlarni V_j to'plamga, qolgan uchlarni esa V_q to'plamga kiritamiz (0 raqamli uch V_j to'plamga kiritiladi). G grafning ikkala uchi ham V_j to'plamga tegishli barcha qirralari kortejini U_j bilan, uning ikkala uchi ham V_q to'plamga tegishli barcha qirralari kortejini esa U_q bilan belgilaymiz. Agar U_j va U_q kortejlar bo'sh bo'lsa, u holda berilgan G graf ikki bo'laklidir, aks holda u ikki bo'lakli emas.

Hozirgacha $k > 2$ bo'lgan hol uchun grafning k bo'lakliligini aniqlash bo'yicha oddiy usul topilmagan.

1.5. Eyler va Gamilton graflari

Eyler graflari. Graflar nazariyasining shakllanishi Kyonigsberg ko'priklari haqidagi masala bilan bog'liq ekanligi yaxshi ma'lum. L. Eyler 1736 yilda bu masalaning yechimga ega emasligini isbotladi. U graflar nazariyasining ancha umumiy hisoblangan quyidagi savoliga ham javob topdi: qanday shartlar bajarilganda bog'lamli grafda barcha qirralardan faqat bir marta o'tadigan sikl mavjud bo'ladi?

Grafning har bir qirrasidan faqat bir marta o'tadigan zanjir **Eyler zanjiri** deb ataladi. Yopiq Eyler zanjiriga (ja'ni **Eyler sikliga**) ega graf **Eyler grafi** deb ataladi. Agar grafda yopiq bo'lmagan Eyler zanjiri topilsa, u holda bunday graf **yarim Eyler grafi** deb ataladi.

1.6- teorema. *Bog'lamli graf Eyler grafi bo'lishi uchun undagi barcha uchlarning darajalari juft bo'lishi zarur va yetarlidir.*

Oriyentirlangan graflarda oriyentirlangan Eyler yo'lini izlash bilan shug'ullanish mumkin. Har bir yoydan faqat bir marta o'tadigan yo'l **oriyentirlangan Eyler yo'li** deb ataladi. Tarkibida oriyentirlangan Eyler yo'li bor bo'lgan oriyentirlangan graf **oriyentirlangan Eyler grafi** deb ataladi.

Endi qirralari soni n ga teng bo'lgan berilgan Eyler grafida Eyler zanjirini tuzishning **Flyori algoritmini** keltiramiz. Bu algoritmgga ko'ra grafning qirralari Eyler siklida uchrashi tartibi bo'yicha 1dan n gacha raqamlab chiqiladi.

Berilgan Eyler grafi uchun Flyori algoritmgga binoan quyidagi ikkita qoida asosida ishlar ketma-ket bajariladi:

1. Grafning ixtiyoriy v uchidan boshlab bu uchga insident bo'lgan istalgan qirraga (masalan, vv' qirraga) 1 raqami beriladi. Bu qirra grafdan olib tashlanadi va v uchdan v' uchga (ya'ni olib tashlangan qirraga insident uchga) o'tiladi.

2. Oxirgi o'tishdan oldingi o'tish natijasida hosil bo'lgan uch w bo'lsin va oxirgi o'tishda biror qirraga k raqami berilgan deylik. w uchga insident istalgan qirra imkoniyati boricha shunday tanlanadiki, bu qirrani olib tashlash grafdagi bog'lamlilikni buzmasin. Tanlangan qirraga navbatdagi $(k+1)$ raqami beriladi va bu qirra grafdan olib tashlanadi. ■

Flyori algoritmiga ko'ra ish yuritish Eyler grafi uchun doimo chekli jarayon ekanligi va bu jarayon doimo grafdan barcha qirralarning olib tashlanishi, ya'ni Eyler zanjirini tuzish bilan tugashi isbotlangan. Shuni ham ta'kidlash kerakki, Flyori algoritmini qo'llash jarayonida qirralarni tanlash imkoniyatlari ko'p bo'lgani uchun, bunday vaziyatlarda, algoritmni qo'llash mavjud Eyler sikllaridan birini topish bilan cheklanadi. Tushunarliki, Flyori algoritmini takror qo'llab (bunda qirralarni tanlash jaroyoni algoritmini avvalgi qo'llashlardagidek aynan takrorlanmasligi kerak) grafda mavjud bo'lgan barcha Eyler sikllarini topish mumkin.

Gamilton graflari. Graflar nazariyasining natijalari muayyan shartlarni qanoatlantiruvchi marshrutlarni topish masalasiga keltiriluvchi bir qator muammolarni hal etishda qo'llanilishi mumkin. Shunday muammolardan biri sifatida Uilyam Gamilton nomi bilan bog'liq masalani keltiramiz. U. Gamilton dodekaedrni tekshirib, uning har bir uchidan faqat bir marta o'tadigan siklni izlab topgan va shu asosda 1859 yilda "Olam bo'ylab sayohat" nomli o'yinni topgan.

Grafning har bir uchidan faqat bir marta o'tadigan zanjir **Gamilton zanjiri** deb ataladi. Yopiq Gamilton zanjiriga (ja'ni **Gamilton sikliga**) ega graf **Gamilton grafi** deb ataladi. Agar grafda yopiq bo'lmagan Gamilton zanjiri topilsa, u holda bunday graf **yarim Gamilton grafi** deb ataladi.

Oriyentirlangan graflarda ham grafning har bir uchidan faqat bir marta o'tuvchi oriyentirlangan sikllarni qarash mumkin

Eyler va Gamilton graflari bir-birlariga o'xshash ta'riflansada, grafning Gamilton grafi ekanligini tasdiqlaydigan alomat (mezon) topish masalasi ancha murakkab muammo hisoblanadi. Hozirgi vaqtgacha graflar nazariyasida grafning

Gamilton grafi ekanligini tasdiqlovchi shartlarni o'rganish bo'yicha izlanishlar davom etib, bu sohadagi ishlar hanuzgacha dolzarbligini yo'qotmasdan kelmoqda.

Qandaydir shartlarga bo'ysunuvchi graflarda Gamilton sikli mavjudligi haqida bir necha tasdiqlar mavjud. Qator hollarda bu tasdiqlarning isbotlari konstruktiv bo'lganligidan, Gamilton siklini tuzishga doir samarali algoritmlar ham yaratilgan.

Berilgan grafda Gamilton zanjirining mavjudligi shartlarni o'rganish bo'yicha izlanishlar davom etayotganligi va bu sohadagi ishlar bugungi kunda ham dolzarbligini yo'qotmaganligi yuqorida qayd etilgan edi. Grafdagi uchlar soni m ning qiymatiga nisbatan ko'phad bilan chegaralangan sondagi qadam ishlatib istalgan grafda Gamilton zanjiri mavjudligini tekshiradigan algoritm hozirgacha topilmagan. Shuning uchun ham Gamilton zanjirini topish masalasi graflar nazariyasida markaziy masalalardan biri bo'lib qolmoqda, Albatta, grafdagi m ta uchlarning $m!$ ta turli ketma-ketliklarini (aniqrog'i, takrorlanmaydigan o'rin almashtirishlarini) tuzib grafda Hamilton zanjiri mavjudligi masalasini hal qilish mumkin. Shunday bo'lishiga qaramasdan, barcha $m!$ ta o'rin almashtirishlarini bajarmasdan qadamlar sonini jiddiy qisqartiradigan umumiy algoritm bor.

Grafda Gamilton zanjirini topish masalasi quyidagi **kommivoyajer masalasi** deb ataluvchi masalada oshkora namoyon bo'ladi. Bir-birlari bilan yo'llar (graf qirralari) vositasida bog'langan shaharlar (graf uchlari) tarmog'i berilgan bo'lib, shaharlarning har bir (a,b) jufti uchun masofa (uni $\mu(a,b)$ bilan belgilaymiz) mos qo'yilgan hamda o'zaro bog'lanmagan shaharlar orasidagi masofa cheksiz katta deb hisoblaymiz. Kommivoyajer uchun shunday Gamilton sikli to-pish kerakki, $\sum_{i=0}^{n-1} \mu(a_i, a_{i+1})$ ifodaning qiymati minimal bo'lsin, bu yerda a_i – tarmoqdagi i - shahar ($i = \overline{0, n}$). Boshqacha aytganda, kommivoyajerning biror shahardan chiqib va qolgan barcha shaharlardan faqat bir martadan o'tib, yana dastlabki shaharga qaytishi imkonini beruvchi eng kichik umumiy uzunlikka ega bolgan yo'lni topish kerak.

1.6. Grafning metrik xarakteristiklari

Graflarda masofa tushunchasi. Bog‘lamli $G=(V,U)$ graf berilgan bo‘lsin. Bu grafda har qanday ikkita v_1 va v_2 uchlar bog‘langan bo‘lgani uchun chetlari v_1 va v_2 uchlardan iborat bo‘lgan hech bo‘lmasa bitta marshrut bor. v_1 va v_2 uchlarni bog‘lovchi eng qisqa (v_1,v_2) marshrutning uzunligi v_1 va v_2 **uchlar orasidagi masofa** deb ataladi va u $d(v_1,v_2)$ bilan belgilanadi. Ravshanki, eng qisqa marshrut oddiy zanjirdir. Tabiiy ravishda $d(v,v)=0$ deb qabul qilamiz.

Shuni ta’kidlash kerakki, graflarda masofa tushunchasini boshqa usul bilan ham aniqlash mumkin.

Yuqoridagi usul bilan aniqlangan masofa **metrika aksiomalari** deb ataluvchi quyidagi shartlarni qanoatlantiradi:

- 1) $d(v_1,v_2) \geq 0$;
- 2) $v_1 = v_2$ bo‘lgandagina $d(v_1,v_2)=0$ bo‘ladi;
- 3) $d(v_1,v_2)=d(v_2,v_1)$;
- 4) $d(v_1,v_2)+d(v_2,v_3) \geq d(v_1,v_3)$.

Oxirgi aksioma **uchburchak tengsizligi** deb ataladi.

Bog‘lamli $G=(V,U)$ graf berilgan bo‘lsin. G grafning ixtiyoriy $v \in V$ uchi uchun aniqlangan

$$e(v) = \max_{w \in V} d(v,w)$$

miqdor shu v **uchning eksentrisiteti** deb ataladi.

Bog‘lamli G graf barcha uchlarining eksentrisitetlari orasidagi qiymati eng kattasi (maksimali) shu **grafning diametri** deb ataladi.

G grafning diametri, odatda, $d(G)$ bilan belgilanadi: $d(G) = \max_{v \in V} e(v)$.

Diametr bu grafning istalgan ikki uchi orasidagi mumkin bo‘lgan eng katta masofadir, ya’ni $d(G) = \max_{v_1,v_2 \in V} d(v_1,v_2)$.

Uzunligi $d(G)$ ga teng bo‘lgan oddiy zanjir **diametral zanjir** deb ataladi.

Tabiiyki, grafda diametral zanjir yagona bo‘lmisligi mumkin.

Bog‘lamli G graf barcha uchlarining eksentrisitetlari orasidagi qiymati eng kichigi (minimali) shu **grafning radiusi** deb ataladi.

G grafning radiusi, odatda, $r(G)$ bilan belgilanadi: $r(G) = \min_{v \in V} e(v)$.

Ravshanki, $r(G) = \min_{v_1 \in V} \max_{v_2 \in V} d(v_1, v_2)$.

Bog‘lamli G grafdagi eksentrisiteti radiusga teng v_0 uch **grafning markazi** (**markaziy uchi**) deb ataladi.

Agar v_0 uch G grafning markazi bo‘lsa, u holda $e(v_0) = \min_{v \in V} e(v)$ bo‘ladi, ya’ni grafning markaziy uchi minimal eksentrisitetga egadir.

Agar grafning markazidan boshqa biror uchigacha bo‘lgan oddiy zanjir eng uzun masofaga ega bo‘lsa, u holda bu zanjir **radial oddiy zanjir** deb ataladi.

Tabiiyki, grafning radiusi uning diametridan katta emas va graf bittadan ko‘p markazga ega bo‘lishi ham mumkin. Bundan tashqari, grafning barcha uchlari uning markaziy uchlari bo‘lishi ham mumkin.

Grafning markaziy uchlarini topish bilan bog‘liq masalalar aholiga xizmat ko‘rsatadigan qandaydir ob‘yektning (kasalxona, maktab va shu kabilarning) joylashish o‘rnini aniqlash bilan bog‘liq muammolarni hal qilishda qo‘llanilishi mumkin. Ta’kidlash kerakki, muayyan vaziyatlarda, ko‘pincha, boshqa holatlarni, jumladan, ob‘yektgacha borish vaqti, punktlar orasidagi masofa va shu kabilarni hisobga olishga to‘g‘ri keladi. Bunday vaziyatlarda **joylashtirishning minimaks masalalari** deb ataluvchi masalalar vujudga keladi.

II BOB. Minimal masofaga ega bo'lgan masalalar va ularni yechish algoritmlari hamda dasturiy vositalari

Ushbu bobda ishga oid bir necha masalalar: minimal masofaga ega bo'lgan yo'l va uning uzunligini aniqlash hamda qiziqarli boshqotirma (kalkulyator) masalasini yechish algoritmlari keltirilib, ular uchun ob'ektga yo'naltirilgan "Delphi" dasturlash tilida, viziullashtirilgan dasturiy vositalari ishlab chiqilgan hamda ulardan foydalanish uchun ko'rsatmalar berilgan.

2.1. Ba'zi ekstremal masalalarni yechishda graflar nazariyasi elementlaridan foydalanish (masalaning qo'yilishi va uni yechish algoritmi)

Biz I bobda keltirilgan graflar nazariyasining asosiy tushuncalaridan foydalanib, quyida **2.1-2.3-masalalarni** yechish algoritmini keltiramiz.

2.1-masala. Boshlang'ich va oxirgi uchlari hamda yoylari uzunligi bilan berilgan uchlari soni N ta bo'lgan graflarda eng qisqa masofaga ega bo'lgan marshrut hamda uning yo'li uzunligini aniqlovchi dastur tuzilsin.

2.2-masala. Uchlari soni N ta bo'lgan graflarda yoylari uzunligi, boshlang'ich va oxirgi uchlarini kiritish orqali berilgan grafdagi eng qisqa yo'l (marshrut) uzunligini aniqlovchi dastur tuzilsin.

2.1-2.2-masalalarni yechish algoritmi. Minimal uzunlikka ega yo'l haqidagi masala. $G=(V,U)$ oriyentirlangan graf berilgan bo'lsin, bu yerda $V=\{1,2,...,m\}$. G grafning biror $s \in V$ uchidan boshqa $t \in V$ uchiga boruvchi yo'llar orasida uzunligi eng kichik bo'lganini topish masalasi bilan shug'ullanamiz. Bu masalani **minimal uzunlikka ega yo'l haqidagi masala** deb ataymiz. Quyida bu masalaning umumlashmasi hisoblangan masalani qarab, uni ham o'sha nom bilan ataymiz.

Grafdagi (i,j) yoyning uzunligini c_{ij} bilan belgilab, $C=(c_{ij})$, $i,j=\overline{1,m}$, matritsa berilgan deb hisoblaymiz. Yuqorida ta'kidlaganlarimizga ko'ra, C

matritsaning c_{ij} elementlari orasida manfiylari yoki nolga tenglari ham bo'lishlari mumkin. Agar grafda biror i uchdan chiqib j uchga kiruvchi yoy mavjud bo'lmasa, u holda bu yoyning uzunligini cheksiz katta deb qabul qilamiz ($c_{ij} = \infty$). Bundan tashqari, G grafda umumiy uzunligi manfiy bo'lgan sikl mavjud emas deb hisoblaymiz, chunki aks holda uzunligi eng kichik bo'lgan yo'l mavjud emas.

Minimal uzunlikka ega yo'l haqidagi masalani hal etish usullari orasida Deykstra tomonidan taklif etilgan algoritm ko'p qo'llaniladi. Quyida grafning 1 belgili uchidan chiqib (bu uchni manba deb qabul qilamiz) grafdagi ixtiyoriy k uchgacha (bu uchni oxirgi uch deb hisoblaymiz) eng qisqa uzunlikka ega yo'lni topish imkonini beruvchi **Deykstra algoritmi** keltirilgan.

Dastlabki qadam. Manbaga (1 belgili uchga) $\varepsilon_1 = 0$ qiymatni mos qo'yib, bu uchni dastlab $R = \emptyset$ deb qabul qilingan R to'plamga kiritamiz: $R = \{1\}$. $\bar{R} = V \setminus R$ deb olamiz.

Umumiy qadam. Boshlang'ich uchi R to'plamga, oxirgi uchi esa $\bar{R}$ to'plamga tegishli bo'lgan barcha yoylar to'plami $(R, \bar{R})$ bo'lsin. Har bir $(i, j) \in (R, \bar{R})$ yoy uchun $h_{ij} = \varepsilon_i + c_{ij}$ miqdorni aniqlaymiz, bu yerda ε_i deb $i \in R$ uchga mos qo'yilgan qiymat (grafning 1 belgili uchidan chiqib i belgili uchigacha eng qisqa yo'l uzunligi) belgilangan.

$\varepsilon_j = \min_{(i,j) \in (R, \bar{R})} h_{ij}$ qiymatni aniqlaymiz. $(R, \bar{R})$ to'plamning oxirgi tenglikda minimum qiymat beruvchi barcha elementlarini, ya'ni (i, j) yoylarni ajratamiz. Ajratilgan yoylarning har biridagi $j \in \bar{R}$ belgili uchga ε_j qiymatni mos qo'yamiz. ε_j qiymat mos qo'yilgan barcha j uchlarni $\bar{R}$ to'plamdan chiqarib R to'plamga kiritamiz.

Ikkala uchi ham R to'plamga tegishli bo'lgan barcha (i, j) yoylar uchun $\varepsilon_i + c_{ij} \geq \varepsilon_j$ tengsizlikning bajarilishini tekshiramiz. Tekshirilayotgan tengsizlik o'rinli bo'lmagan (ja'ni $\varepsilon_{j_*} > \varepsilon_i + c_{ij_*}$ bo'lgan) barcha j_* belgili uchlarning har biriga mos qo'yilgan eski ε_{j_*} qiymat o'rniga yangi $\varepsilon_i + c_{ij_*}$ qiymatni mos qo'yamiz va

(i, j_*) yoyni ajratamiz. Bunda eski ε_{j_*} qiymat aniqlangan paytda ajratilgan yoyni ajratilmagan deb hisoblaymiz.

Uchlarga qiymat mos qo'yish jarayonini oxirgi (k belgili) uchga qiymat mos qo'yilguncha davom ettiramiz. Grafning 1 belgili uchidan (manbadan) chiqib uning ixtiyoriy k uchigacha (oxirgi uchigacha) eng qisqa yo'l uzunligi ε_k bo'ladi.

Oxirgi qadam. Grafning oxirgi uchidan boshlab ajratilgan yoylar yo'nalishiga qarama-qarshi yo'nalishda uning 1 belgili uchiga kelguncha harakatlanib, natijada grafdagi 1 belgili uchdan ixtiyoriy k uchgacha eng qisqa uzunlikka ega yo'l(lar)ni topamiz. ■

Bu dasturiy vositalardan foydalanish natijasida xususiy hollar uchun natijalar quyidagilardan iborat.

2.1-masalada boshlanishi 1-uch oxiri 5-uch yoylari uzunliklari esa 1-uchdan 2- uchgacha bo'lgan masofa 15, 2-uchdan 3- uchgacha bo'lgan masofa 10, 2-uchdan 4- uchgacha bo'lgan masofa 8, 3-uchdan 1- uchgacha bo'lgan masofa 12, 3-uchdan 5- uchgacha bo'lgan masofa 4, 4-uchdan 5- uchgacha bo'lgan masofa 8 ga teng bo'lgan grafda eng qisqa masofaga ega bo'lgan marshrut 1-3-5 hamda uning yo'li uzunligi 19 ga teng bo'lishi aniqlandi (2.3-chizmaga qarang).

2.3-masalada 25 soni uchun masala shartiga ko'ra 5 ta amal bajarilar;

479 soni uchun masala shartiga ko'ra 11 ta amal bajarilar ekan.

2.2. Dasturdan foydalanuvchilar uchun ko'rsatmalar

Biz bu bandda 2.1-2.3-masalalarni keltirib, ularni yechish uchun ishlab chiqilgan dasturdan foydalanuvchilarga ko'rsatmalar berib o'tamiz.

2.1-masala. Boshlang'ich va oxirgi uchlari hamda yoylari uzunligi bilan berilgan uchlari soni N ta bo'lgan graflarda eng qisqa masofaga ega bo'lgan marshrut hamda uning yo'li uzunligini aniqlovchi dastur tuzilsin.

2.2-masala. Uchlari soni N ta bo'lgan graflarda yoylari uzunligi, boshlang'ich va oxirgi uchlarini kiritish orqali berilgan grafdagi eng qisqa yo'l (marshrut) uzunligini aniqlovchi dastur tuzilsin.

2.3-masala (Kalkulyator). Quyidagi amallarni bajaruvchi kalkulyator berilgan:

X sonini 2 ga ko'paytirish;

X sonini 3 ga ko'paytirish;

X soniga birni qo'shish;

1 sonidan N sonini hosil qilish uchun kamida nechta amal bajarish kerak.

Kiruvchi ma'lumotlar:

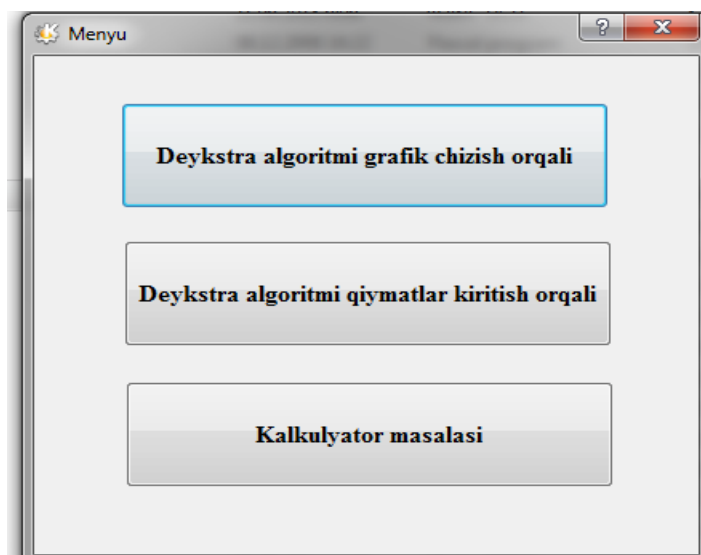
1000000 dan oshmaydigan N soni berilgan.

Chiquvchi ma'lumotlar:

Birinchi satrda minimal amallar sonini chiqaring.

Yuqorida keltirilgan masalalarni yechish uchun dasturdan quyidagicha foydalaniladi.

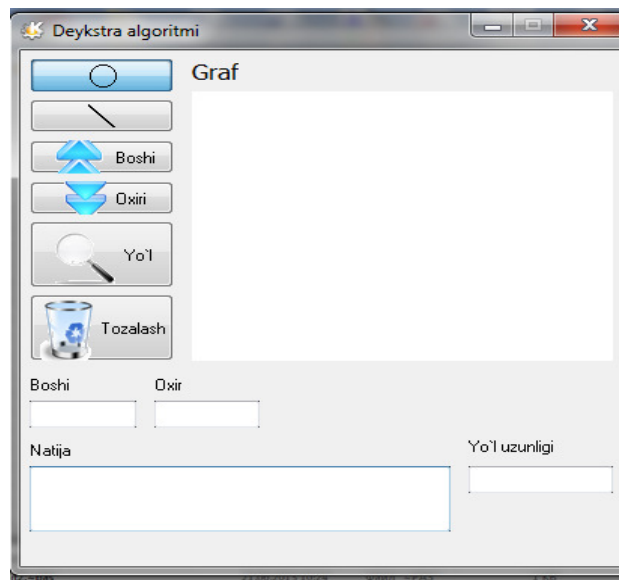
1. Deystra.exe fayli ishga tushiriladi. Natijada quyidagi dastur oynasi hosil bo'ladi:



2.1-rasm. Dastur bosh oynasi

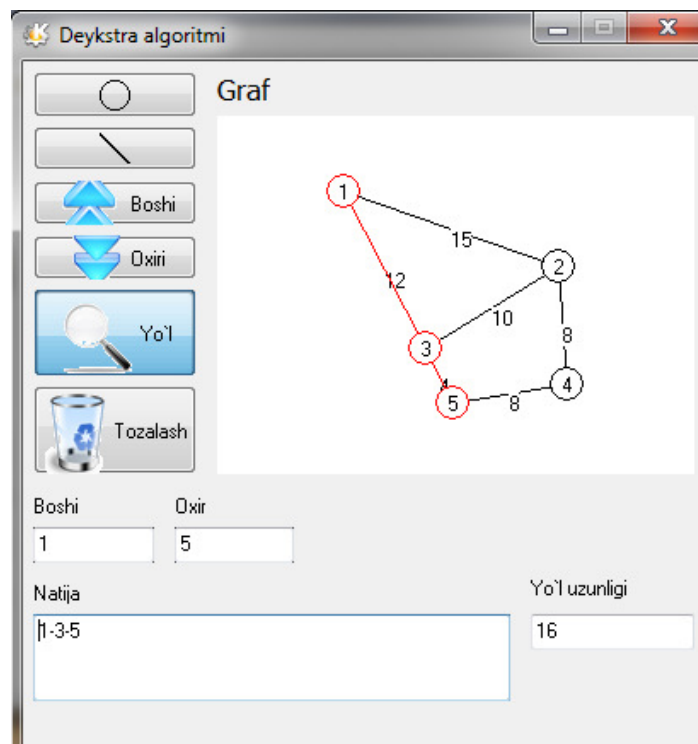
2. Bu oynada quyidagi tugmalar joylashgan:

2.1. Deysktra algoritmi grafik chizish orqali;



2.2-rasm. Deysktra algoritmi grafik chizish orqali;

Bunda dastlab oynaning chap qismida joylashgan aylana shaklidagi tugma tanlanadi va grafni hosil qilish maydonining tegishli (ixtiyoriy) joylariga sichqonchaning chap tugmasini bir martadan bosiladi, natijada maydonda grafning 1-,2-,...,n-uchlari hosil bo'ladi. Keyin uchlarni tutashtiruvchi to'g'ri chiziq belgili tugma tanlanadi va tegishli (tutashtirilishi lozim bo'lgan) uchlar bu tugma yordamida o'zaro tutashtiriladi, natijada uchlar orasidagi masofalar hosil bo'ladi unda masofalar dastur yordamida to'g'ridan-to'g'ri aniqlanadi (2.2-rasmga qar.).



2.3-rasm. Nyuton binomini hisoblash oynasi

Hosil qilingan grafning boshlang'ich va oxirgi uchlari mos holda "boshi" va "oxiri" nuqtalari yordamida tanladi. So'ngra "Yo'l" tugmasi bosiladi unda chizilgan grafning eng qisqa marshruti qizil rangda ajratib ko'rsatiladi, "Natija" va "Yo'l uzunligi" maydonida qaralayotgan grafning eng qisqa marshruti va bu marshrutning yo'li uzunligi hosil bo'ladi. "Tozalash" tugmasini bosish orqali kiritilgan qiymatlar va natijalar aks etgan maydonlar tozalanadi. Xuddi shu tarzda boshqa hosil qilinib, ular uchun yuqoridagi amallar takrorlanadi.

Xulosa

Mazkur bitiruv malakaviy ishida turli minimal masofaga ega bo'lgan yo'lni va uning uzunligini aniqlash hamda qiziqarli boshqotirma (kalkulyator) masalasi masalalari qaraldi va ularni yechishda ob'ektga yo'naltirilgan "Delphi" dasturlash tilida qulay interfeysga ega bo'lgan (viziullashtirilgan) dasturiy vositalari yaratildi hamda ulardan quyidagi natijalar olindi:

2.1-masala. Boshlang'ich va oxirgi uchlari hamda yoylari uzunligi bilan berilgan uchlari soni N ta bo'lgan graflarda eng qisqa masofaga ega bo'lgan marshrut hamda uning yo'li uzunligini aniqlovchi dastur tuzilsin.

2.2-masala. Uchlari soni N ta bo'lgan graflarda yoylari uzunligi, boshlang'ich va oxirgi uchlarini kiritish orqali berilgan grafdagi eng qisqa yo'l (marshrut) uzunligini aniqlovchi dastur tuzilsin.

2.3-masala (Kalkulyator). Quyidagi amallarni bajaruvchi kalkulyator berilgan:

X sonini 2 ga ko'paytirish;

X sonini 3 ga ko'paytirish;

X soniga birni qo'shish;

1 sonidan N sonini hosil qilish uchun kamida nechta amal bajarish kerak.

Kiruvchi ma'lumotlar:

1000000 dan oshmaydigan N soni berilgan.

Chiquvchi ma'lumotlar:

Birinchi satrda minimal amallar sonini chiqaring.

Bu masalalarni yechish uchun ob'ektga yo'naltirilgan "Delphi" dasturlash tilida qulay interfeysga ega bo'lgan viziullashtirilgan dasturiy vositalar ishlab chiqildi (1-3-illovalarga qarang).

Bu dasturiy vositalardan foydalanish natijasida:

2.1-masalada boshlanishi 1-uch oxiri 5-uch yoylari uzunliklari esa 1-uchdan 2- uchgacha bo'lgan masofa 15, 2-uchdan 3- uchgacha bo'lgan masofa 10, 2-uchdan 4- uchgacha bo'lgan masofa 8, 3-uchdan 1- uchgacha bo'lgan masofa 12,

3-uchdan 5- uchgacha bo'lgan masofa 4, 4-uchdan 5- uchgacha bo'lgan masofa 8 ga teng bo'lgan grafda eng qisqa masofaga ega bo'lgan marshrut 1-3-5 hamda uning yo'li uzunligi 19 ga teng bo'lishi aniqlandi (2.3-chizmaga qarang).

2.3-masalada 25 soni uchun masala shartiga ko'ra 5 ta amal bajarilar;

479 soni uchun masala shartiga ko'ra 11 ta amal bajarilar ekan.

Foydalanilgan adabiyotlar

1. Окулов С. М. Программирование в алгоритмах.— М.: БИНОМ. Лаборатория знаний, 2004. — 341 с: ил.
2. Головешкин В. А., Ульянов М. В. Теория рекурсии для программистов. М.: ФИЗМАТЛИТ, 2006. 296 с.
3. Ахо, Альфред, В., Хопкрофт, Джон, Ульман, Джеффри, Д. Структуры данных и алгоритмы. : Пер. с англ. : Уч. пос. — М. : Издательский дом "Вильямс", 2000. — 384 с. : ил. — Парал. тит. англ. (рус).
4. Ф.А. Новиков Дискретная математика для программистов. СПб: Питер, 2000. — 304 с.
5. Иванов Б.Н. Дискретная математика. Алгоритмы и программы: Учеб. пособие / Лаборатория Базовых Знаний, 2003. —288 с.
6. To'rayev N.T., Azizov I., Otaqulov S. Kombinatorika va graflar nazariyasi. Uslubiy qo'llanma: Samarqand. 2006. — 262 b.
7. В. Гофман, А. Хоманенко. Delphi 7. — СПб.: БХВ–Петербург, 2004 г.
8. Дарахвелидзе П. Г., Марков Е. П. Д20 Программирование в Delphi 7. — СПб.: БХВ-Петербург, 2003. — 784 с : ил.
9. Краснов М. В. OpenGL. Графика в проектах Delphi. — СПб.: БХВ-Петербург, 2002. — 352 с: ил.
10. <http://olo.looblogs.info/issledovanie-funkcij-maple.html>
11. <http://maple.plusby.com/index.html>

Ilova

1-ilo va.

```
unit Unit1;
interface
    uses
        Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
        Dialogs,
        Buttons, ExtCtrls, StdCtrls, XPMAN, Menus;
    type
        TForm1 = class(TForm)
            SpeedButton1: TSpeedButton;
            SpeedButton2: TSpeedButton;
            SpeedButton3: TSpeedButton;
            SpeedButton4: TSpeedButton;
            SpeedButton5: TSpeedButton;
            SpeedButton6: TSpeedButton;
            Memo1: TMemo;
            Label1: TLabel;
            Label2: TLabel;
            Edit1: TEdit;
            Label3: TLabel;
            Edit2: TEdit;
            Edit3: TEdit;
            Label4: TLabel;
            XPManifest1: TXPManifest;
            Image: TImage;
            Label5: TLabel;
            procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
                Y: Integer);
            procedure SpeedButton1Click(Sender: TObject);
            procedure ImageMouseUp(Sender: TObject; Button: TMouseButton;
                Shift: TShiftState; X, Y: Integer);
            procedure ImageMouseMove(Sender: TObject; Shift: TShiftState; X,
                Y: Integer);
            procedure ImageMouseDown(Sender: TObject; Button: TMouseButton;
                Shift: TShiftState; X, Y: Integer);
            procedure SpeedButton2Click(Sender: TObject);
```

```

    procedure FormCreate(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
    procedure SpeedButton6Click(Sender: TObject);
    procedure APropos2Click(Sender: TObject);
    procedure FormCenter;
  private
    Drawing: Boolean;
    Origin, MovePt: TPoint;
    DrawingTool : byte;
    { Private declarations }
  public
    procedure DrawShape(TopLeft, BottomRight: TPoint; AMode:
TPenMode);
    { Public declarations }
  end;
var
  Form1: TForm1;
  implementation
{$R *.DFM}
uses disktrat, Unit2;
    const max = 30;
  type Vertek = record
    posx,posy : Integer;
  end;
    AVertek = Array [1..max] of Vertek;
    var count : byte;
    awal,akhir : byte;
    Node : AVertek;
    bool_awal,bool_akhir : Boolean;
    node1,node2 : byte;
    Data : TJarak;
    Closed : TPath;
    procedure TForm1.DrawShape(TopLeft, BottomRight: TPoint;
AMode: TPenMode);
  begin
    with Image.Canvas do
      begin

```

```

    Pen.Mode := AMode;
case DrawingTool of
  2: {LINE}
  begin
    Image.Canvas.MoveTo(TopLeft.X, TopLeft.Y);
    Image.Canvas.LineTo(BottomRight.X, BottomRight.Y);
  end;
end;
end;
end;
    procedure TForm1.FormMouseMove(Sender: TObject; Shift:
TShiftState; X,
Y: Integer);
begin
  if Drawing then
  begin
    DrawShape(Origin, MovePt, pmNotXor);
    MovePt := Point(X, Y);
    DrawShape(Origin, MovePt, pmNotXor);
  end;
end;
    procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
  Drawingtool := 1;
  Image.Canvas.Pen.Mode := pmcopy;
  end;
  procedure TForm1.ImageMouseUp(Sender: TObject; Button:
TMouseButton;
Shift: TShiftState; X, Y: Integer);
var XX,YY,i : byte;
begin
  if Drawing then
  begin
    DrawShape(Origin, MovePt, pmNotXor); //afficher;
    if drawingtool=2 then
    begin
      bool_akhir := False;
      for i := 1 to count do

```

```

        if (X>Node[i].posX-10) and (Y>Node[i].posY-10) and
(X<Node[i].posX+10) and (Y<Node[i].posY+10) then
begin
    bool_akhir := True;
    node2 := i;
    break;
end;
        if (node1<>0) and (node2<>0) and bool_awal and bool_akhir then
begin
    DrawShape(Point(Node[node1].posx,Node[node1].posy),
Point(Node[node2].posx,Node[node2].posy), pmCopy);
    Data[node1,node2] := round(sqrt(sqrt(abs(Node[node2].posy-
Node[node1].posy)/9) + sqrt(abs(Node[node2].posx-Node[node1].posx)/9)));
    Data[node2,node1] := Data[node1,node2];
    XX := Node[node1].posx;
    YY := Node[node1].posy;
    with Image.Canvas do
begin
    Image.Canvas.Pen.Mode := pmcopy;
    Ellipse(XX-10,YY-10,XX+10,YY+10);
    if node1 div 10 > 0 then
        Textout(xX-7,Yy-6,IntToStr(node1))
    else
        Textout(Xx-3,Yy-6,IntToStr(node1));
end;
    XX := Node[node2].posx;
    YY := Node[node2].posy;

    with Image.Canvas do
begin
    Image.Canvas.Pen.Mode := pmcopy;
    Ellipse(XX-10,YY-10,XX+10,YY+10);
    if node2 div 10 > 0 then
        Textout(xX-7,Yy-6,IntToStr(node2))
    else
        Textout(Xx-3,Yy-6,IntToStr(node2));
end;

```



```

        image.Canvas.TextOut((Node[node1].posx+Node[node2].posx)div
2, (Node[node1].posy+Node[node2].posy) div
2,IntToStr(Data[node1,node2]));
        end;
    end; //if
    Drawing := False;
        if drawingtool=1 then
    begin
        count := count + 1;
        with Node[count] do
        begin
            posx := x;
            posy := y;
        end;
        with Image.Canvas do
        begin
            Image.Canvas.Pen.Mode := pmcopy;
            Ellipse(X-10,Y-10,X+10,Y+10);
            if count div 10 > 0 then
                Textout(x-7,y-6,IntToStr(count))
            else
                Textout(x-3,y-6,IntToStr(count));
            end;
        end;
    end;
end;
end;
        procedure TForm1.ImageMouseMove(Sender: TObject; Shift:
TShiftState; X,
Y: Integer);
begin
    if Drawing then
    begin
        DrawShape(Origin, MovePt, pmNotXor);
        MovePt := Point(X, Y);
        DrawShape(Origin, MovePt, pmNotXor);
    end;
end;
        procedure TForm1.ImageMouseDown(Sender: TObject; Button:
TMouseButton;

```

```

Shift: TShiftState; X, Y: Integer);
var i : byte;
begin
  Drawing := True;
  Image.Canvas.MoveTo(X, Y);
  Origin := Point(X, Y);
  MovePt := Origin;
  if drawingtool=2 then
  begin
    bool_awal := False;
    for i := 1 to count do
      if (X>Node[i].posX-10) and (Y>Node[i].posY-10) and
(X<Node[i].posX+10) and (Y<Node[i].posY+10) then
        begin
          bool_awal := True;
          node1 := i;
          break;
        end;
      end
    else
      if drawingtool in [3,4] then
      begin
        for i := 1 to count do
          if (X>Node[i].posX-10) and (Y>Node[i].posY-10) and
(X<Node[i].posX+10) and (Y<Node[i].posY+10) then
            begin
              case drawingtool of
                3 : begin
                  awal := i;
                  edit2.Text := IntToStr(i);
                end;
                4 : begin
                  akhir := i;
                  edit3.Text := IntToStr(i);
                end;
              end; //case
            break;
          end; //if
        end;
      end;
    end;
  end;

```

```

end;
    procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
    Drawingtool := 2;
    end;
procedure TForm1.FormCreate(Sender: TObject);
var
    Bitmap: TBitmap;
    xx,yy : byte;
begin
    Form1.Left := Screen.Width div 2 - Width div 2;
    Form1.Top := Screen.Height div 2 - Height div 2;
    DoubleBuffered := True;
    Bitmap := nil;
    try
        Bitmap := TBitmap.Create;
        Bitmap.Width := 350;
        Bitmap.Height := 300;
        Image.Picture.Graphic := Bitmap;
    finally
        Bitmap.Free;
    end;
    Memo1.Clear;
    Drawingtool := 1;
    count := 0;
    awal:= 0;
    akhir := 0;
    edit2.Text := "";
    edit3.Text := "";
    for xx := 1 to max do
        for yy := 1 to max do
            begin
                if xx=yy then
                    Data[xx,yy] := 0
                else
                    Data[xx,yy] := 999;
                end;
            end;
        end;
    end;
    procedure TForm1.SpeedButton3Click(Sender: TObject);

```

```

begin
    Drawingtool := 3;
end;

procedure TForm1.SpeedButton4Click(Sender: TObject);
begin
    Drawingtool := 4;
end;

procedure TForm1.SpeedButton5Click(Sender: TObject);
var i : byte;
    XX,YY : byte;
begin
    memo1.Clear;
    Drawingtool := 2;
    RuteTerpendek(Data,Closed,awal,akhir,count);
    if (awal<>0) and (akhir<>0) and (closed.jarak<>0) and
(closed.jarak<>999) then
        begin
            Drawing := True;
            edit1.Text := IntToStr(closed.jarak);
            for i := 1 to closed.nodeke-1 do
                begin
                    memo1.Text := memo1.Text + IntToStr(closed.arraypath[i]) + '-';
                    image.Canvas.Pen.Color := clred;
                    DrawShape(Point(Node[closed.arraypath[i]].posx,Node[closed.arraypath[i]].
posy), Point(Node[closed.arraypath[i+1]].posx,Node[closed.arraypath[i+1]].posy),
pmCopy);
                    XX := Node[closed.arraypath[i]].posx;
                    YY := Node[closed.arraypath[i]].posy;
                    with Image.Canvas do
                        begin
                            Image.Canvas.Pen.Mode := pmcopy;
                            Ellipse(XX-10,YY-10,XX+10,YY+10);
                            if node2 div 10 > 0 then
                                Textout(xX-7,Yy-6,IntToStr(closed.arraypath[i]))
                            else
                                Textout(Xx-3,Yy-6,IntToStr(closed.arraypath[i]));
                        end;
                    end;
                    XX := Node[closed.arraypath[closed.nodeke]].posx;

```

```

YY := Node[closed.arraypath[closed.nodeke]].posy;
with Image.Canvas do
begin
  Image.Canvas.Pen.Mode := pmcopy;
  Ellipse(XX-10,YY-10,XX+10,YY+10);
  if closed.nodeke div 10 > 0 then
    Textout(xX-7,Yy-6,IntToStr(closed.arraypath[closed.nodeke]))
  else
    Textout(Xx-3,Yy-6,IntToStr(closed.arraypath[closed.nodeke]));
end;
image.Canvas.Pen.Color := clblack;
memo1.Text := memo1.Text +
IntToStr(closed.arraypath[closed.nodeke]);
  Drawing := False;
end
else
begin
  memo1.Text := 'Yo`llar bir-biri bilan bog`lanmagan';
  edit1.Text := '';
end;
Drawingtool := 5;
end;
  procedure TForm1.SpeedButton6Click(Sender: TObject);
begin
  Drawingtool := 6;
  FormCreate(Sender);
  SpeedButton1Click(Sender);
  edit1.Clear;
end;
  procedure TForm1.FormCenter;
begin
  with Form2 do begin
    Left := Screen.Width div 2 - Width div 2;
    Top := Screen.Height div 2 - Height div 2;
  end;
end;
  procedure TForm1.APropos2Click(Sender: TObject);
begin
  with Form2 do begin

```

```

Form2 := TForm2.Create(Application);
{ Centre la fiche }
FormCenter;
{ Application de la fonction : AnimateWindow }
AnimateWindow(Handle, 250{ Vitesse}, AW_CENTER);
Show;
end;
end;
end.

```

2-ilova.

```

unit Unit5;
interface
    uses
        Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
        Dialogs, StdCtrls, Grids;
type
    TForm5 = class(TForm)
        Button1: TButton;
        StringGrid1: TStringGrid;
        ListBox1: TListBox;
        Edit1: TEdit;
        Edit2: TEdit;
        Label1: TLabel;
        Label2: TLabel;
        Label3: TLabel;
        Label4: TLabel;
        Label5: TLabel;
        Button2: TButton;
        Edit3: TEdit;
        Button3: TButton;
        Button4: TButton;
        procedure Button1Click(Sender: TObject);
        procedure FormCreate(Sender: TObject);
        procedure Button3Click(Sender: TObject);
        procedure Button4Click(Sender: TObject);
        procedure Button2Click(Sender: TObject);
    private

```

```

    { Private declarations }
public
    { Public declarations }
end;

var
Form5: TForm5;
n:integer;
implementation
    {$R *.dfm}
//hisoblash jarayoni
procedure TForm5.Button1Click(Sender: TObject);
var
    a:array[1..50,1..50] of longint;
    b:array[1..50]of boolean;
    d:array[1..50] of longint;
    q, i, j, m, v: integer;
begin
    q := StrToIntDef(Edit1.Text, 1);
    if (q < 1) or (q > n) then q := 1;
    for i := 1 to n do
    for j := 1 to n do
    a[j, i] := StrToIntDef(StringGrid1.Cells[i - 1, j - 1], -1);
    fillchar(b,sizeof(b),0);
    fillchar(d,sizeof(d), 10000);
    d[q] := 0;
    for i:=1 to n do
    begin
    m:=1000;
    for j:=1 to n do
    if ( (d[j] <= m) and (not b[j]) ) then
    begin
    m:=d[j];
    v:=j;
    end;
    b[v] := true;
    for j:=1 to n do
    if ((a[v,j]<>-1)and(not b[j])and (d[v]+a[v,j]<d[j])) then
    begin
    d[j]:=d[v]+a[v,j];

```

```

    end;
end;
    ListBox1.Clear;
for i := 1 to n do
if ((i=strtoint(Edit2.text))
and
(q=strtoint(edit1.text))) then
    ListBox1.Items.Append('Boshi: '+IntToStr(q) + ' - Oxiri: ' + IntToStr(i) + '
Yo`l uzunligi: ' + IntToStr(d[i]));
    end;
    //boshlang`ich qiymarlar berish
procedure TForm5.FormCreate(Sender: TObject);
    var i, j, fuck: integer;
begin
    n:=7;
    StringGrid1.RowCount:=n;
    StringGrid1.ColCount:=n;
    for i := 0 to n - 1 do StringGrid1.Cells[i, i] := '0';
    for i := 0 to n - 1 do
    for j := i + 1 to n - 1 do
    begin
    StringGrid1.Cells[i, j] := '0';
    StringGrid1.Cells[j, i] := StringGrid1.Cells[i, j]
    end;
    end;
    //tozalash
procedure TForm5.Button3Click(Sender: TObject);
var i, j, fuck: integer;
begin
    n:=StrToInt(Edit3.Text);
    StringGrid1.RowCount:=n;
    StringGrid1.ColCount:=n;
    for i := 0 to n - 1 do StringGrid1.Cells[i, i] := '0';
    for i := 0 to n - 1 do
    for j := i + 1 to n - 1 do
    begin
    StringGrid1.Cells[i, j] := '0';
    StringGrid1.Cells[j, i] := StringGrid1.Cells[i, j]
    end;

```



```

end;
//tasodifiy qiymarlar bilan to`ldirish
procedure TForm5.Button4Click(Sender: TObject);
var i, j, fuck: integer;
begin
    n:=StrToInt(Edit3.Text);
    StringGrid1.RowCount:=n;
    StringGrid1.ColCount:=n;
    Randomize;
    for i := 0 to n - 1 do StringGrid1.Cells[i, i] := '0';
    for i := 0 to n - 1 do
        for j := i + 1 to n - 1 do
            begin
                StringGrid1.Cells[i, j] := IntToStr(random(10));
                if StringGrid1.Cells[i, j] = '0' then StringGrid1.Cells[i, j] := '-1';
                StringGrid1.Cells[j, i] := StringGrid1.Cells[i, j]
            end;
        end;
    end;

    //tozalsh funksiyasi
    procedure TForm5.Button2Click(Sender: TObject);
    var i, j, fuck: integer;
    begin
        n:=7;
        StringGrid1.RowCount:=n;
        StringGrid1.ColCount:=n;
        for i := 0 to n - 1 do StringGrid1.Cells[i, i] := '0';
        for i := 0 to n - 1 do
            for j := i + 1 to n - 1 do
                begin
                    StringGrid1.Cells[i, j] := '0';
                    StringGrid1.Cells[j, i] := StringGrid1.Cells[i, j]
                end;
            end;
        Edit1.Text:="";
        Edit2.Text:="";
        Edit3.Text:="";
        ListBox1.Clear;
    end;
end.

```

```

3-ilova.
unit Unit3;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, StdCtrls, ComCtrls;
type
  TForm3 = class(TForm)
    RichEdit1: TRichEdit;
    Label1: TLabel;
    Edit1: TEdit;
    Label2: TLabel;
    Label3: TLabel;
    Edit2: TEdit;
    Button1: TButton;
    Button2: TButton;
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;
  d:array[1..1000006] of integer;
n,i:integer;
implementation
{$R *.dfm}
  //tozalash
procedure TForm3.Button2Click(Sender: TObject);
begin
  Edit1.Text:="";
  Edit2.Text:=""
end;
  //hisoblash
procedure TForm3.Button1Click(Sender: TObject);

```

```

begin
  n:=StrToInt(Edit1.Text);
  d[1]:= 0;
  d[2]:= 1;
  d[3]:= 1;
  for i:=3 to n do
  begin
    d[i]:= d[i - 1] + 1;
    if (i mod 2 = 0) then
    begin
      if (d[i] > d[i div 2] + 1) then
        d[i]:= d[i div 2] + 1;
      end;
      if (i mod 3 = 0) then
      begin
        if (d[i] > d[i div 3] + 1) then
          d[i]:= d[i div 3] + 1;
        end;
      end;
    end;
  end;
  Edit2.Text:=IntToStr(d[n]);
end;
end.

```