

**O‘ZBEKISTON RESPUBLIKASI OLIY VA O‘RTA MAXSUS TA’LIM
VAZIRLIGI**

ALISHER NAVOIY NOMIDAGI SAMARQAND DAVLAT UNIVERSITETI

**MEXANIKA-MATEMATIKA
FAKULTETI**

«MATEMATIK MODEL SHTIRISH» KAFEDRASI

**«MUAYYAN OB’YEKTLARNING GRAF MODELINI HOSIL QILISH VA
ULARNI TAHLIL QILISHNING DASTURIY VOSITALARINI
YARATISH» mavzusidagi**

BITIRUV MALAKAVIY ISHI

Bajardi: «Amaliy matematika va
informatika» ta’lim yo‘nalishi
bitiruvchi 4 kurs talabasi
TOSHTEMIROV JONIBEK
MUXIDINOVICH.

Ilmiy rahbar:
dots. BOZOROV I.N.

Bitiruv malakaviy ishi kafedradan dastlabki himoyadan o‘tdi.

_____ sonli bayonnomasi «_____» _____ 2014 yil.

SAMARQAND – 2014 y.

MUNDARIJA

	KIRISH.....	3
1-BOB	Graflar nazariyasining asosiy tushunchalari.....	6
1.1	Graflar nazariyasining dastlabki ma'lumotlari.....	6
1.2	Graflarning berilish usullari va ular ustida amallar.....	11
1.3	Marshrutlar va zanjirlar. Eyler va Gamilton graflari.....	18
1.4	Graflarning metrik xarakteristikalarini aniqlash algoritmi va dasturiy vositalari.....	24
2-BOB	Muayyan ob'ektlarning graf modelini hosil qilish algoritmi va dasturiy vositalari.....	27
2.1	Berilgan uchlar qo'shniligi va insidentlik matritsalariga mos graflarni hosil qilish algoritmi va dasturiy vositalari.....	27
2.2	Hosil qilingan graflarni Eyler va Gamilton sikllariga tashkirlash algoritmi va dasturiy vositalari.....	29
2.3	Graflarning metrik xarakteristikalarini aniqlash algoritmi va dasturiy vositalari.....	33
2.4	Dasturdan foydalanuvchilar uchun ko'rsatmalar.....	35
	XULOSA.....	42
	Foydalanilgan adabiyotlar.....	43
	ILOVALAR	

Kirish

Masalaning qo'yilishi. Muayyan ob'yektlarning graf modelini uchlar qo'shniligi va insidentlik matritsalarini yordamida hosil qilish. Hosil qilingan modellarni tahlil qilish algoritmi va dasturiy vositalarini ishlab chiqish.

Mavzuning dolzarbligi. Hozirgi vaqtda "Graflar nazariyasi" bo'yicha tadqiqotlar natijalari inson faoliyatining turli sohalarida qo'llaniladi. Ulardan ba'zilar quyidagilardir: boshqotirmalarni hal qilish; qiziqarli o'yinlar; yo'llar (transport harakatlarini boshqarish sohasida), elektr zanjirlari, integral sxemalari va boshqarish tizimlarini loyihalashtirish; avtomatlar, blok-sxemalar va komp'yuter uchun dasturlarni tadqiq qilish va hokazo.

Shu sababli ushbu ishda muayyan ob'yektlarning graf modeli hosil qilinadi va hosil qilingan modellarni tahlil qilish algoritmi va dasturiy vositalarini ishlab chiqish masalalari qaratiladi.

Ishning maqsad va vazifalari. Bitiruv malakaviy ishning asosiy maqsadi uchlar qo'shniligi va insidentlik matritsalarini yordamida muayyan ob'yektlarning graf modelini hosil qilish va hosil qilingan modellarni tahlil qilish uchun ob'ektga yo'naltirilgan "Delphi" dasturlash tilida viziullashtirilgan dasturiy vositalarini ishlab chiqishdan iborat.

Ilmiy-tadqiqot usullari. Ushbu bitiruv malakaviy ishida muayyan ob'yektlarning graf modelini hosil qilish uchun uchlar qo'shniligi va insidentlik matritsalarini orqali qurilgan grafning radiusi, diametri, markazi hamda Eyler va Gamilton graflari sikli va yo'lini aniqlash 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.

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 ahamiytga ega bo‘lgan diskret matematikaning bir tilidir.

Tadqiqotning ilmiy yangiligi. Bitiruv malakaviy ishida olingan natijalar amaliy-uslubiy xarakterga ega bo‘lib, ishda sirtmoqqa va karrali qirralarga ega bo‘lmagan uchlar qo‘shniligi va insidentlik matritsalarini yordamida muayyan ob‘yektlarning graf modelini hosil qilish hamda hosil qilingan modellarni tahlil qilish uchun ob‘ektga yo‘naltirilgan “Delphi” dasturlash tilida viziullashtirilgan dasturiy vositalari ishlab chiqilgan.

Tadqiqot predmeti va ob‘ekti. Tadqiqotning predmeti “Graflar nazariyasi”, “Axborot xavfsizligi”, “Kompyuter injineriingi”, “Dastur injineriingi” va shu kabi fan sohalari bo‘lib, ob‘ekti esa muayyan ob‘ektlarning graf modellaridan iborat.

Tadqiqotning ilmiy va amaliy ahamiyati. Ishda olingan natijalar va unda qo‘llanilgan usullardan turli iqtisodiy, ijtimoiy sohalarning ko‘pgina amaliy masalalarining graf modellarini hosil qilishda, ularning dasturiy vositalarini ishlab chiqishda, “Graflar nazariyasi”, “Axborot xavfsizligi”, “Kompyuter injineriingi”, “Dastur injineriingi” va shu kabi fanlarning amaliy mashg‘ulotlari o‘quv jarayonlarida dasturiy vosita sifatida foydalanish mumkin.

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

I bob 4 ta banddan iborat bo‘lib, unda olingan natijalarni bayon qilishda zarur bo‘lgan asosiy tushunchalar (ta’rif va tasdiqlar): 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, Eyler va Gamilton graflari haqida ma’lumotlar berilgan.

II bobda ishga oid muayyan ob’ektlarning graf modelarini uchlar qo‘shniligi, insidentlik matritsalari orqali hosil qilish. Hosil qilingan graflarni Eyler va Gamilton graflariga tekshirish, ularning metrik xarakteristikalari: uchlar orasidagi masofa matritsasi, graf radiusi, diametri, markazlarini aniqlashga oid masalalar qaralib, bu masalalarni yechish uchun ob’ektga yo‘naltirilgan “Delphi” dasturlash tilida, viziullashtirilgan dasturiy vosita ishlab chiqilgan va dasturdan foydalanuvchilar uchun ko‘rsatmalar berilgan.

Olingan natijalarning qisqacha mazmuni. Bitiruv malakaviy ishida muayyan ob’ektlarning graf modelini uchlar qo‘shniligi va insidentlik matritsalari yordamida hosil qilish. Hosil qilingan graf modellarini Eyler va Gamilton sikliga tekshirish, graflarning metrik xarakteristikalarini: uchlar orasidagi masofa matritsasi, graf radiusi, diametri, markazlarini aniqlash algoritmi tuzildi va vizuallashtirilgan dasturiy vosita ishlab chiqildi. Bu dasturiy vositadan foydalanib, uchlar qo‘shniligi va insidentlik matritsalari yordamida turli graflarni hosil qilish va ularni Eyler va Gamilton sikliga tekshirish, graflarning metrik xarakteristikalarini: uchlar orasidagi masofa matritsasi, graf radiusi, diametri, markazlarini aniqlash bo‘yicha bir necha masalalar ko‘rib chilgan (qar. 2.1-2.3-masalalar).

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 hamda graflarning metrik xarakteristikalarini haqida ma'lumotlar berilgan.

1.1. Graflar nazariyasining dastlabki 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'plamining 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 $(\overline{a,b})$ yoki $(\overline{a},b)$, qirra uchun $(\overline{a},\overline{b})$, 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 qirralardan iborat bo'lsa, u holda **yo'naltirilmagan (oriyentirlanmagan)** va faqat yo'naltirilgan (oriyentirlangan) qirralardan (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 qirralari ham, oriyentirlangan qirralari 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) korteji 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'pdur, 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, $1, 2, \dots, m$ 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 ularga mos bo'lgan $x', y' \in V'$ ($x \leftrightarrow y, x' \leftrightarrow y'$) 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 tasdiqsi) ushbu bobning 1.4- bandida keltirilgan.

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

1.2. Graflarning berilish usullari va ular ustida amallar

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 xossalarni 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 uchlarini tekislikda yoki fazoda nuqtalar bilan, qirralarini (yoylarini) esa mos uchlarini tutashtiruvchi uzluksiz chiziqlar bilan ifodalab, qandaydir diagrammaga – **grafning ko'rgazmali tasviriga** ega bo'lamiz. Agar uchlar to'plami va bu uchlarining 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 uchlarini 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 mukinligi 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-tasdiq. *Har qanday chekli grafni 3 o'lchovli Evklid fazosida geometrik ifodalash mumkin.*

Shuni ham ta'kidlash kerakki, 1- tasdiqdagi 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.

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.

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

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 oriyentirlanmagan multigraf bo'lsin. a_{ij} elementlari G grafning i va j uchlari tutashtiruvchi qirralar soniga

teng bo'lgan $A = (a_{ij})$ ($i, j = 1, 2, \dots, m$) matritsa **oriyentirlanmagan multigrafning uchlari qo'shnili matritsasi** deb ataladi.

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

1.2- tasdiq. *Graflar faqat va faqat uchlari qo'shnili matritsalarini 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'shnili 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 yakkaqat 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'shnili matritsasi** deb ataladi.

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

Insidentlik matritsalarini. 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 uchlari satrlari, qirralariga esa ustunlari mos keluvchi va elementlari

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

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

Graflar ustida sodda amallar. Graflar ustida turli amallar bajarish mumkin, masalan, graflarni birlashtirish, biriktirish, 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 qirranga (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 uchlaridan iborat bo'lgan shunday yagona $\overline{G}$ graf mavjudki, $\overline{G}$ grafdagi barcha juft uchlar 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 uchlariga 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 uchlariga insident u_1 qirra hamda v va v_2 uchlariga 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:

– uchlari (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 $v'' = (v_1'', v_2'') \in V$ uchlari faqat va faqat shu holda qo'shni bo'lakilarki, 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.

1.3. Marshrutlar va zanjirlar. Eyler va Gamilton graflari

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);
- boshlang'ich uchga ega bo'lib, oxirgi uchga ega bo'lmasligi mumkin yoki, aksincha, oxirgi uchga ega bo'lib, boshlang'ich 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- tasdiq. 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‘g‘langan 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 grafdagi **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- tasdiq (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-tasdiqning tasdig‘idagi $m+r=2+n$ tenglik **Eylar formulasi** deb ataladi.

Eyler tasdiqsidan bir qator natijalar kelib chiqadi. Masalan, bu tasdiqdan 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‘prik** 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.

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 Eyer zanjiriga (ja'ni **Eyler sikliga**) ega graf **Eyler grafi** deb ataladi. Agar grafda yopiq bo'lmagan Eyer zanjiri topilsa, u holda bunday graf **yarim Eyer grafi** deb ataladi.

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

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

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

Berilgan Eyer grafi uchun Flyori algoritmgaga 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 algoritmgaga ko'ra ish yuritish Eyer grafi uchun doimo chekli jarayon ekanligi va bu jarayon doimo grafdan barcha qirralarning olib tashlanishi, ya'ni Eyer 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 Eyer 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 algoritim bor.

Grafda Hamilton 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 Hamilton 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.

1.4. Grafning metrik xarakteristikalar

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 ekssentrisiteti** deb ataladi.

Bog'lamli G graf barcha uchlarining ekssentrisitetlari 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'lmashligi mumkin.

Bog'lamli G graf barcha uchlarining ekssentrisitetlari 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 ekssentrisiteti 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 ekssentrisitetga 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. Muayyan ob'ektlarning graf modelini hosil qilish algoritmi va dasturiy vositalari

Ushbu bobda ishga berilgan uchlar qo'shniligi va insidentlik matritsalariga mos graflarni hosil qilish, hosil qilingan graflarni Eyler va Gamilton sikllariga tashkirlash va metrik xarakteristikalarini aniqlash 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. Berilgan uchlar qo'shniligi va insidentlik matritsalariga mos graflarni hosil qilish algoritmi va dasturiy vositalari

Graflarning turlicha berilish usullari mavjud. Grafning abstrakt matematik ta'rifi uning berilish usullaridan biridir. Grafning abstrakt matematik ta'rifi uni tasavvur qilish, anglash, uning xossalarni 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.

Shu sababli biz bu bandda I bobning 2-bandida ko'rib chiqilgan berilgan uchlar qo'shniligi va insidentlik matritsalariga mos graflarning tasvirini hosil qilish algoritmi va dasturiy vositalarini ishlab chiqish uchun quyidagi masalani qaraymiz.

2.1-masala. Berilishi usullariga kg'ra

- a) uchlar qg'shniligi matritsasiga mos grafni hosil qilish;
- b) insidentlik matritsasiga mos grafni hosil qilish dasturi tuzilsin.

Qg'yilgan masalani yechish uchun quyidagi algoritmlar ishlab chiqildi:

a) uchlar qg'shniligi matritsasiga mos grafni hosil qilish

```
procedure TForm1.hosilNuqta(n:integer);  
  var x,y,i,r:integer;  
      da:double;  
begin
```

```

x:=Image1.Width div 2;
y:=Image1.Height div 2;
r:=150;
da:=2*pi/n;
for i:=1 to n+1 do
begin
arr[i].x:=x+round(r*cos(pi/2+i*da));
arr[i].y:=y-round(r*sin(pi/2+i*da));
end;
end;
procedure TForm1.StringGridtoMatrix;
var i,j,n,m:integer;
begin
n:=StringGrid1.RowCount-1;
m:=StringGrid1.ColCount-1;
for i:=1 to n do
for j:=1 to m do
A[i,j]:=StrToInt(StringGrid1.Cells[j,i]);
end;
end;
procedure TForm1.MatrixtoStringGrid;
var i,j,n,m:integer;
begin
n:=StringGrid3.RowCount-1;
for i:=1 to n do
for j:=1 to n do
StringGrid1.Cells[j,i]:=IntToStr(A[i,j]);
end;
end;
procedure TForm1.chizishChiziq( i,j:integer);
var p1,p2:TPoint;
begin
Image1.Canvas.Pen.Style:=psDot;
Image1.Canvas.Pen.Width:=4;
Image1.Canvas.Pen.Color:=clBlue;
Image1.Canvas.MoveTo(arr[j].X,arr[j].Y);
Image1.Canvas.LineTo(arr[i].X,arr[i].Y);
end;
procedure TForm1.chizishGraf;
var i,j,n:integer;
begin
Image1.Canvas.Pen.Color:=clBlack;
Image1.Canvas.Pen.Width:=4;
Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
n:=StringGrid1.RowCount-1;
hosilNuqta(n);
StringGridtoMatrix;
for i:=1 to n do
for j:=i to n do
begin
if(A[i,j]=1) then chizishChiziq(i,j);
end;
Image1.Canvas.Pen.Color:=clBlue;
for i:=1 to n do

```

```

begin
  Image1.Canvas.Ellipse(arr[i].X-10,arr[i].Y-10,arr[i].X+10,arr[i].Y+10);
  Image1.Canvas.TextOut(arr[i].X-5,arr[i].Y-5,IntToStr(i));
end;
end;

```

b) insidentlik matritsasiga mos grafni hosil qilish. Bunda insidentlik matritsasi uchlar qg'shnilik matritsasiga g'tkaziladi va a) band bg'yicha graf hosil qilinadi.

```

procedure TForm1.InsedenttoQushnilik;
var i,j,i1,j1,l,m:integer;
begin
  m:=StringGrid3.RowCount-1;
  l:=StringGrid3.ColCount-1;
  StringGrid3toMatrix;
  for i:=1 to 100 do
    for j:=1 to 100 do
      A[i,j]:=0;
    for i:=1 to l do
      begin
        i1:=0; j1:=0;
        for j:=1 to m do
          if(Ins[j,i]=1) then
            begin
              if(i1>0) then
                j1:=j
              else
                i1:=j;
            end;
          if(i1+j1>1) then
            begin
              A[i1,j1]:=1;
              A[j1,i1]:=1;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

2.2. Hosil qilingan graflarni Eyler va Gamilton sikllariga teshkirish algoritmi va dasturiy vositalari

Kg'pgina amaliy masalalar graf orqali yechishda har bir uchda bir marta g'tish yoki qirradan bir martadan masalalar orqali yechiladi. Bunday masalalarni yechimini topish uchun Eyler va Gamilton sikllariga tekshirish orqali amalga oshiriladi. Shu maqsadni amalga oshirish uhcun Eyler va Gamilton sikllariga

tekshirish algoritmi va dasturiy vositasini ishlab chiqish uchun quyidagi masalalarni qaraymiz.

2.2. masala. Hosil qilingan grafni

- a) Eyler sikliga tekshirish va sikl mavjud bo'lsa, bu siklni topish;
- b) Gamilton sikliga tekshirish va sikl mavjud bo'lsa, bu siklni topish algoritmi va dasturi tuzilsin.

Quyilgan masalani yechish uchun quyidagi algoritmlar ishlab chiqildi:

a) Eyler sikliga tekshirish va sikl mavjud bo'lsa, bu siklni topish.

```
procedure TForm1.EylerSikli;
const
  maxn = 100;
var
  e, was: array[1..maxn, 1..maxn] of integer;
  ne: array[1..maxn] of integer;
  stack: array[1..maxn*maxn] of integer;
  j, i, n, u, v, top: longint;
  ok: boolean;
begin
  n := StringGrid1.RowCount - 1;
  // m := 0;
  StringGrid1.Matrix;
  for i := 1 to n + 1 do
    begin
      ne[i] := 0;
      for j := 1 to n + 1 do
        begin
          e[i, j] := 0;
          stack[(n + 1) * (i - 1) + j] := 0;
          was[i, j] := 0;
        end;
      end;
    for i := 1 to n do
      for j := i to n do
        begin
          if (A[i, j] = 1) then
            begin
              u := i;
              v := j;
              inc(ne[u]); e[u, ne[u]] := v;
              inc(ne[v]); e[v, ne[v]] := u;
            end;
          end;
        j := 0;
      for i := 1 to n do
        begin
          if ne[i] = 0 then inc(j);
```

```

    if ne[i] mod 2=1 then
        begin
            RichEdit1.Text:='Berilgan grafda Eyler sikli mavjud emas.';
            exit;
        end;
    end;
    if j=n then
        begin
            RichEdit1.Text:='Berilgan graf bir-biri bilan bog'lanmagan.';
            exit;
        end;
    top:=1; stack[1]:=1;
    j:=0;
    for i:=1 to 100 do
        eys[i]:=0;
    while top>0 do begin
        u:=stack[top]; ok:=true;
        for i:=1 to ne[u] do
            begin
                v:=e[u,i];
                if was[u,v]=0 then
                    begin
                        inc(top); stack[top]:=v;
                        was[u,v]:=1;was[v,u]:=1;
                        ok:=false; break;
                    end;
            end;
        if ok then
            begin
                dec(top);
                inc(j);
                eys[j]:=u;
            end;
        end;
    chizishGrafEyler(j);
end;
procedure TForm1.chizishGrafEyler(n:integer);
var i:integer;
begin
    RichEdit1.Text:='Eyler sikli yg'li:  ';
    for i:=2 to n do
        begin
            chizishChiziqEyler(eys[i-1],eys[i]);
            RichEdit1.Text:=RichEdit1.Text+IntToStr(eys[i-1])+'->';
        end;
    RichEdit1.Text:=RichEdit1.Text+IntToStr(eys[n]);
    for i:=1 to StringGrid1.RowCount-1 do
        begin
            Image1.Canvas.Pen.Color:=clBlue;
            Image1.Canvas.Ellipse(arr[i].X-10,arr[i].Y-10,arr[i].X+10,arr[i].Y+10);
            Image1.Canvas.TextOut(arr[i].X-5,arr[i].Y-5,IntToStr(i));
        end;
end;

```

end;

b)Gamilton sikliga tekshirish va sikl mavjud bg'lsa, bu siklni topish.

```
procedure TForm1.print_gamilton_c(n:integer);
var
  p,i:integer;
begin
  RichEdit1.Text:='Gamilton sikli yg'li: ';
  for p:=1 to n-1 do
  begin
    chizishChiziqEyler(path[p-1]+1,path[p]+1);
    RichEdit1.Text:=RichEdit1.Text+ IntToStr(path[p-1]+1)+'->';
  end;
  chizishChiziqEyler(path[n-1]+1,path[0]+1);
  RichEdit1.Text:=RichEdit1.Text+IntToStr(path[n-1]+1)+'-
>' +IntToStr(path[0]+1)+' ';
  for i:=1 to StringGrid1.RowCount-1 do
  begin
    Image1.Canvas.Pen.Color:=clBlue;
    Image1.Canvas.Ellipse(arr[i].X-10,arr[i].Y-10,arr[i].X+10,arr[i].Y+10);
    Image1.Canvas.TextOut(arr[i].X-5,arr[i].Y-5,IntToStr(i));
  end;

end;

end;
function TForm1.gamilton(k:integer):integer;
var
  v,gl,n:integer;
begin
  n:=StringGrid1.RowCount-1;
  gl:=0;
  v:=-1;
  while ((v<n) and (gl=0)) do begin
    inc(v);
    if (a[v+1,path[k-1]+1]>0) or (a[path[k-1]+1,v+1]>0) then begin
      if (k=n) and (v=v0) then gl:=1
      else if (c[v]=-1) then begin
        c[v]:=k;
        path[k]:=v;
        gl:=gamilton(k+1);
        if (gl=0) then c[v]:=-1;
      end else continue;
    end;
  end;
  result:=gl;
end;
procedure TForm1.Button4Click(Sender: TObject);
var j,n:integer;
begin
  Button2Click(Self);
  n:=StringGrid1.RowCount-1;
  for j:=0 to n-1 do c[j]:=-1;
```



```

path[0]:=v0;
c[v0]:=v0;
if (gamilton(1)>0) then print_gamilton_c(n) else
RichEdit1.Text:='Gamilton sikli berilgan grafda mavjud emas.';
end;

```

2.3. Graflarning metrik xarakteristikalarini aniqlash algoritmi va dasturiy vositalari

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 graf markazini, radiusi, diametri va masofa matritsasini topish masalalari vujudga keladi.

Shu maqsadlarni amalga oshirish uchun bu bandda grafning metrik xarakteristikalarini aniqlash algoritmi va dasturiy vositasini ishlab chiqish uchun quyidagi masalalar qaraymiz.

2.3-masala. Hosil qilingan graflar uchun:

- a) graf uchlarning bir-biriga yo‘nalishini ifodalaydigan masofa matritsasini aniqlash;
- b) topilgan masofa matritsasi orqali grafning metrik xarakteristikalarini graf radiusi, diametri va markazini aniqlash dasturi tuzilsin.

Qg‘yilgan masalani yechish uchun quyidagi algoritmlar ishlab chiqildi:

a) Masofa matritsasini topish

```

procedure TForm1.matritsaD;
var i,j,n,m:integer;
begin
n:=StringGrid1.RowCount-1;
for i:=1 to n do
for j:=1 to n do
begin
D[i,j]:=A[i,j];
if(D[i,j]=0) then
D[i,j]:=100000;

```

```

        end;
    for i:=1 to n do
        D[i,i]:=0;
    for m:=1 to n do
        for i:=1 to n do
            for j:=1 to n do
                D[i,j]:=min(D[i,j],D[i,m]+D[m,j]);
                StringGrid2.RowCount:=n+1;
                StringGrid2.ColCount:=n+1;
            end;
        end;
    for i:=1 to n do
        for j:=1 to n do
            begin
                if(D[i,j]=100000) then
                    StringGrid2.Cells[i,j]:='infy'
                else
                    StringGrid2.Cells[i,j]:=IntToStr(D[i,j]);
                end;
            end;
        end;
    end;
end;

```

b) graf radiusi, diametri va markazi aniqlash

```

procedure TForm1.maxDElement;
var i,n:integer;
begin
    n:=StringGrid1.RowCount-1;
    for i:=1 to n do
        W[i]:=MaxIntValue(D[i]);
    end;
function TForm1.GrafRadius:integer;
var i,n,min1:integer;
begin
    n:=StringGrid1.RowCount-1;
    min1:=W[1];
    for i:=2 to n do
        min1:=min(min1,W[i]);
        result:=min1;
    end;
function TForm1.GrafDiametr:integer;
var i,n,max1:integer;
begin
    n:=StringGrid1.RowCount-1;
    max1:=W[1];
    for i:=2 to n do
        max1:=max(max1,W[i]);
        result:=max1;
    end;
procedure TForm1.BitBtn2Click(Sender: TObject);
var i,n,gr:integer;
begin
    Button2Click(Self);
    matritsaD;
    maxDElement;
    gr:=GrafRadius;

```

```

if(gr=100000) then
  RichEdit1.Text:='Graf bog‘lamli emas'
else
begin
  RichEdit1.Text:='Graf radiusi: '+IntToStr(GrafRadius)+' ga';
  RichEdit1.Text:=RichEdit1.Text+', diametri: '+IntToStr(GrafDiametr)+' ga
teng.';

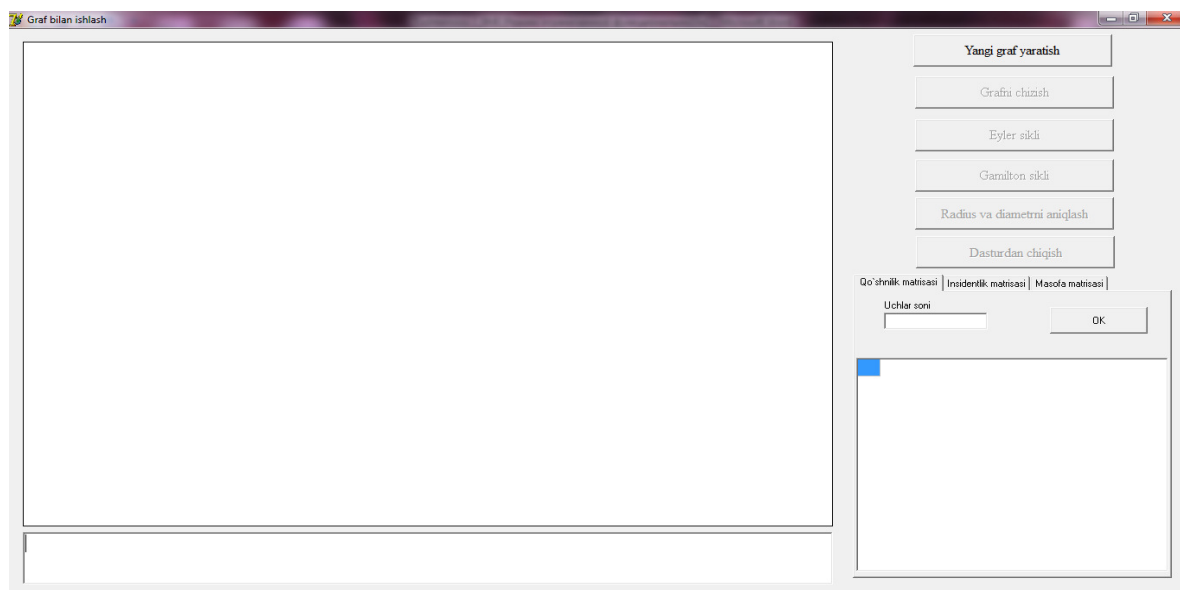
  n:=StringGrid1.RowCount-1;
  RichEdit1.Text:=RichEdit1.Text+#13+'Graf markazi uchlari: ';
  for i:=1 to n do
    if(W[i]=gr) then
      RichEdit1.Text:=RichEdit1.Text+IntToStr(i)+' ';
  end;
end;
end;

```

2.4. Dasturdan foydalanuvchilar uchun ko‘rsatmalar

Biz bu bandda 2.1-2.3-bandlarda keltirilgan Berilgan uchlar qo‘shniligi va insidentlik matritsalariga mos graflarni hosil qilish , hosil qilingan graflarni Eyler va Gamilton sikllariga tashkirlash va graflarning metrik xarakteristikalarini (radius, diametr, graf markazi, masofa matritsasi) ni namoish qilish uchun ishlab chiqilgan viziullashtirilgan dasturiy vositadan foydalanuvchilarga ko‘rsatmalar berib o‘tamiz.

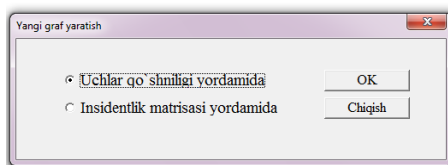
Muayyan ob‘yektlarning graf modelini hosil qilish va ularni tahlil qilishning dasturiy vositasidan foydalanish uchun “graf.exe” fayli ishga tushiriladi. Natijada quyidagi dasturiy vosita oynasi hosil bo‘ladi.



2.1-rasm. Dastur bosh oynasi

2.1-masalaning a) bandini yechish va undan olinadigan natijalarini hosil qilish bo'yicha ko'rsatmalar.

“Yangi graf yaratish” tugmasi bosiladi va uni natijasida 2.2-rasmdagi oyna hosil bo'g'ladi.



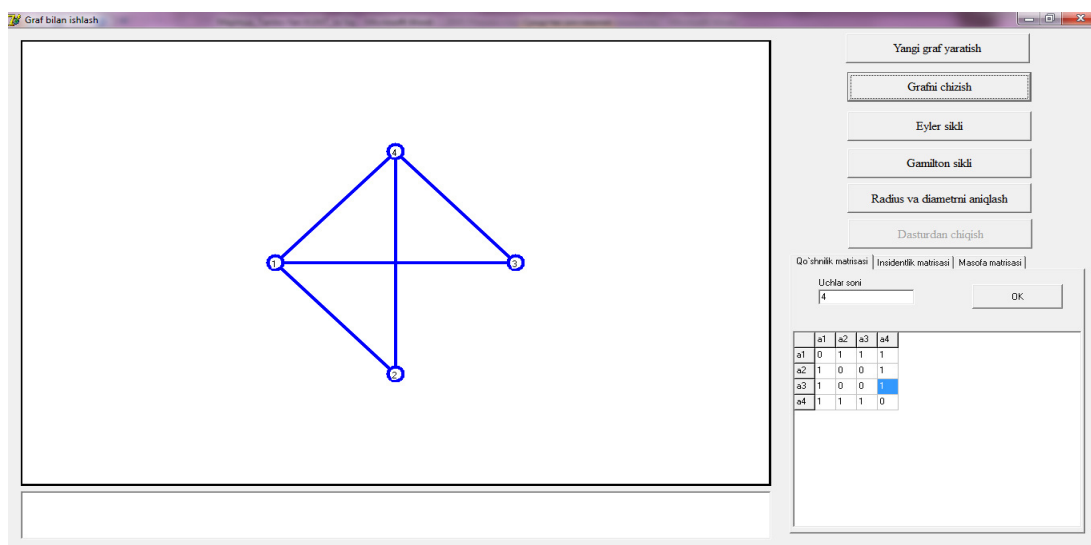
2.2-rasm. Yangi graf yaratish oynasi

2.2-rasmda ko'rsatilgan oynadan “uchlar qo'shniligi yordamida” bandi tanlanib, OK tugmasi bosiladi. “Qo'shnilik matritsasi” bo'limida joylashgan “uchlar soni” kiritish maydoniga grafning uchlar soni kiritiladi va shu bo'limdagi OK tugmasi bosiladi. Natijada shunga mos jadvalli kiritish maydoni hosil qiladi. Jadvalli kiritish maydonida berilgan uchlar qo'shniligi matritsasi tuziladi va grafni hosil qilish uchun “Grafni chizish” tugmasi bosiladi.

Masalan, yuqorida keltirilgan algoritmdan foydalanib, quyidagi grafning uchlari qo'shniligi matritsasi

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

yordamida hosil qilingan grafni 2.3-rasmda ko'rish mumkin.



2.3-rasm. Uchlar qo'shniligi matritsasi yordamida hosil qilingan graf tasviri

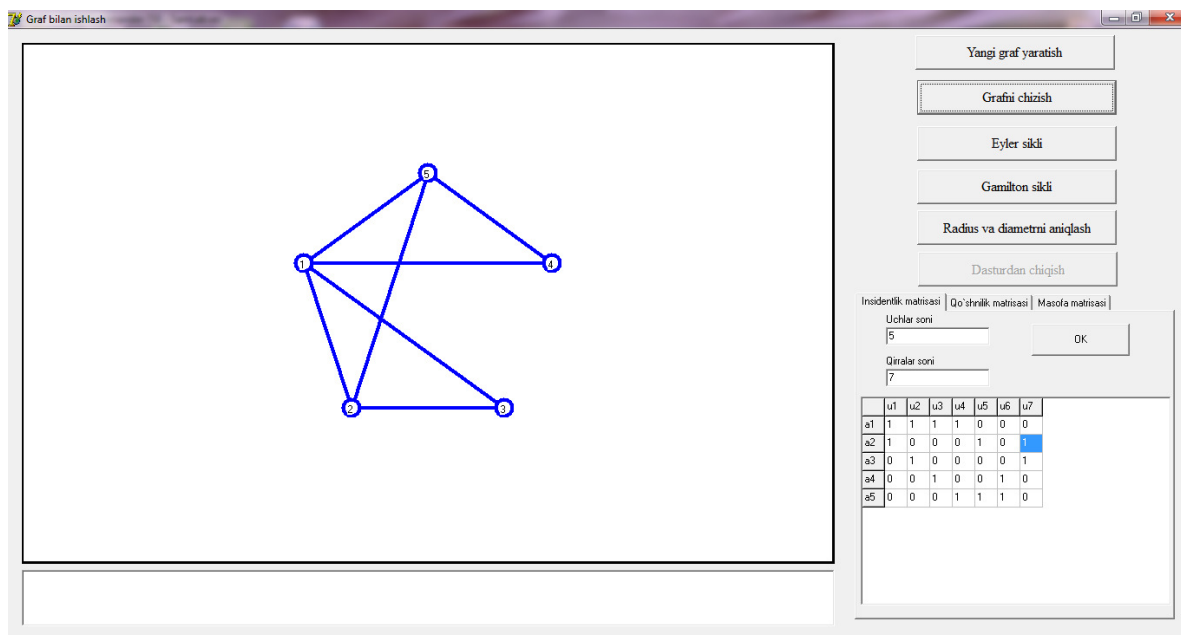
b) bandini yechish va undan olinadigan natijalarini hosil qilish bo'yicha ko'rsatmalar.

“Yangi graf yaratish” tugmasi bosiladi va uni natijasida 2.2-rasmdagi oyna hosil bo'ladi. 2.2-rasmda ko'rsatilgan oynadan “Insidentlik matritsasi yordamida” bandi tanlanib, OK tugmasi bosiladi. “Insidentlik matritsasi” bo'limida joylashgan “uchlar soni” va “qirralar soni” kiritish maydonlariga grafning uchlar va qirralar soni kiritiladi va shu bo'limdagi OK tugmasi bosiladi. Natijada shunga mos jadvalni kiritish maydoni hosil qiladi. Jadvalni kiritish maydonida berilgan insidentlik matritsasi tug'ldiriladi va grafni hosil qilish uchun “Grafni chizish” tugmasi bosiladi.

Masalan, yuqorida keltirilgan algoritmdan foydalanib, quyidagi grafning insidentlik matritsasi

$$B = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

yordamida hosil qilingan grafni 2.4-rasmda ko'rish mumkin.

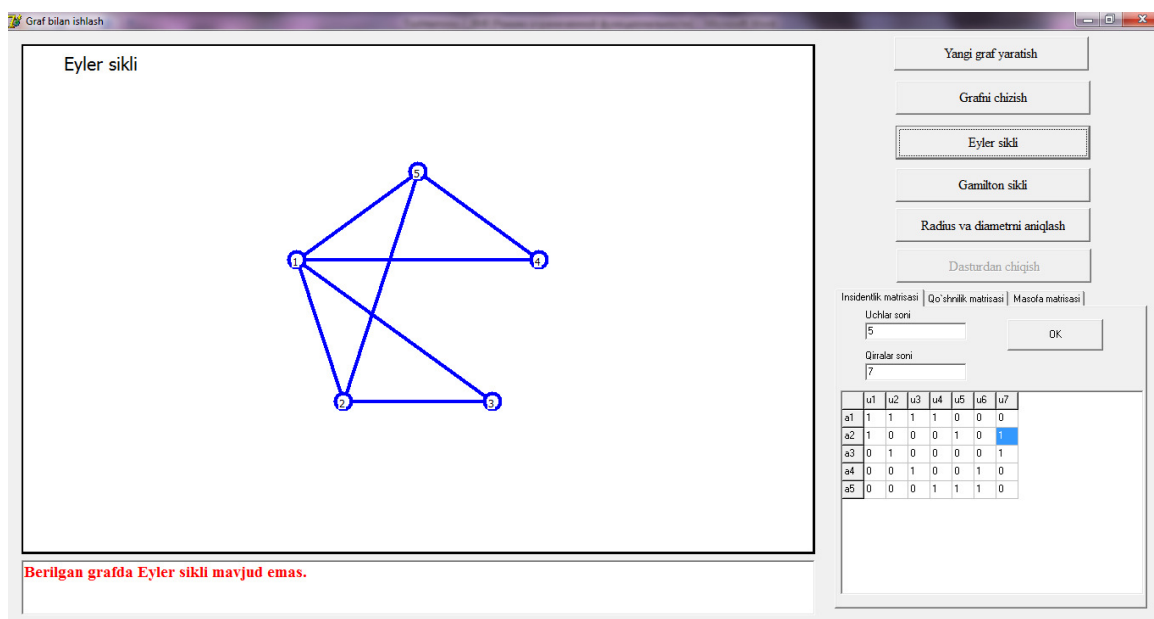


2.4-rasm. Insidentlik matritsasi yordamida hosil qilingan graf tasviri

2.2-masalani a) bandini yechish va undan olinadigan natijalarini hosil qilish bo'yicha ko'rsatmalar.

“Uchlar qg'shniligi” yoki “insidentlik” matritsasi yordamida hosil qilingan grafni Eyler sikliga tekshirish va mavjud bg'lsa bu siklni kg'rsatish uchun yuqoridagi kg'rsatmadagi 2.1.masalani a) yoki b) bandi dasturiy vositada bajarilgandan sg'ng, “Eyler sikli” tugmasi bosiladi. Dasturiy vosita bajarish natijasida ikki holat bg'lishi mumkin: 1-holat Eyler sikli mavjud emas(xabar oynasida “Berilgan grafda Eyler sikli mavjud emas” chiqariladi), 2-holat Eyler sikli mavjud va siklni aniqlash va kg'rsatish(xabar oynasida “Eyler sikli yg'li: dastur topgan yg'l” chiqariladi).

Masalan, 1-holat uchun 2.1-masalada b) bandidagi kg'rilgan misol uchun qurilgan grafni dasturiy vosita orqali Eyler sikliga tekshirish natijasi quyidagicha(2.5-rasm).

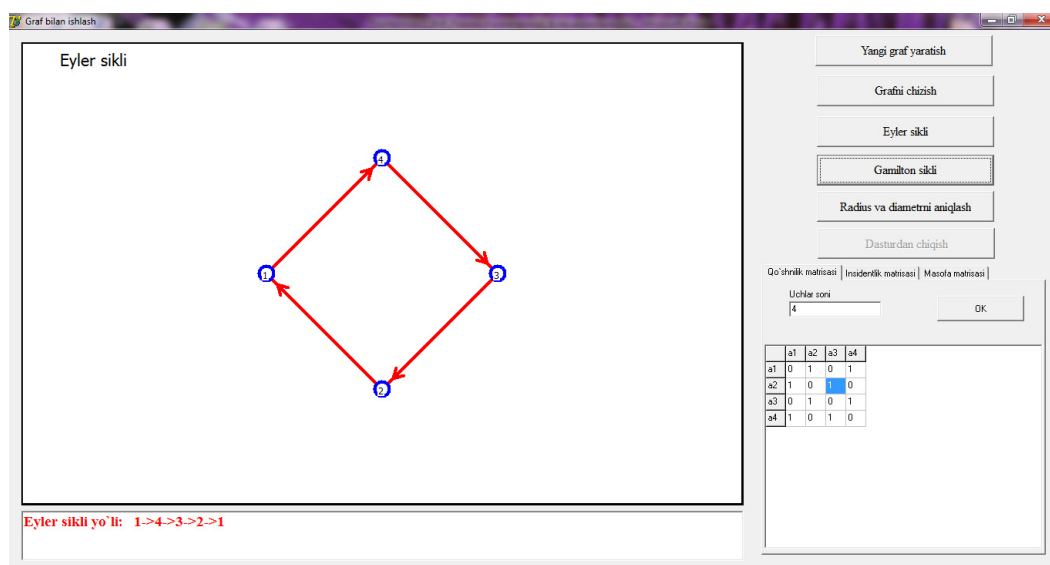


2.5-rasm. Hosil qilingan grafni Eyler sikli tekshirishda mavjud bg'lmagan holati.

2-holat uchun Dasturiy vosita orqali quyidagi “uchlar qg'shniligi” matritsasi orqali hosil qilingan

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

grafni Eyler sikliga tekshirish natijasi quyidagicha(2.6-rasm).



2.6-rasm. Hosil qilingan grafni Eylar sikli tekshirishda mavjud bg'lgan holati.

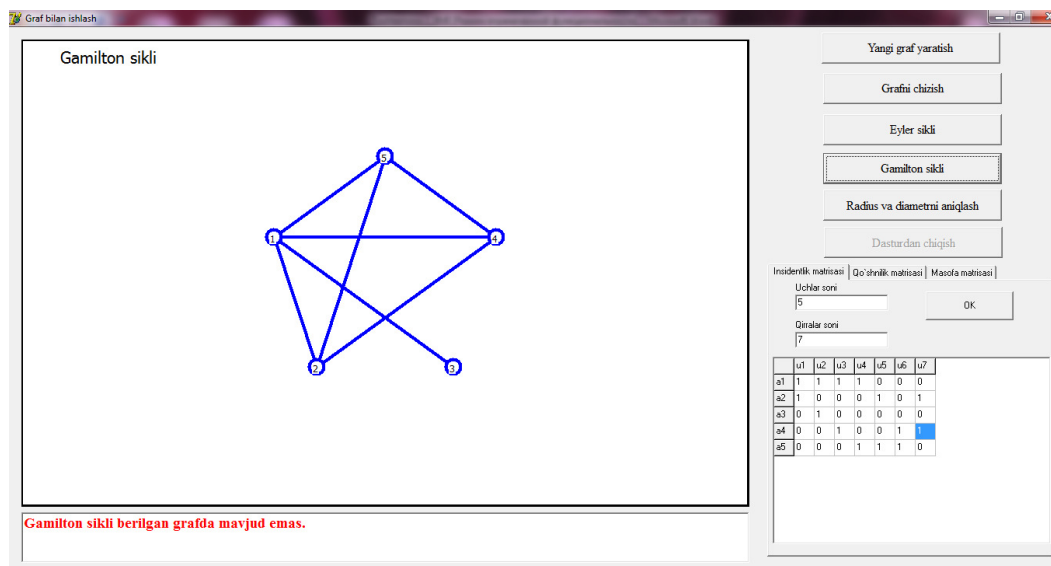
b) bandini yechish va undan olinadigan natijalarini hosil qilish bo'yicha ko'rsatmalar.

“Uchlar qg'shniligi” yoki “insidentlik” matritsasi yordamida hosil qilingan grafni Gamilton sikliga tekshirish va mavjud bg'lsa bu siklni kg'rsatish uchun yuqoridagi kg'rsatmadagi 2.1.masalani a) yoki b) bandi dasturiy vositada bajarilgandan sg'ng, “Gamilton sikli” tugmasi bosiladi. Dasturiy vosita bajarish natijasida ikki holat bg'lishi mumkin: 1-holat Gamilton sikli mavjud emas(xabar oynasida “Berilgan grafda Gamilton sikli mavjud emas” chiqariladi), 2-holat Gamilton sikli mavjud va siklni aniqlash va kg'rsatish(xabar oynasida “Gamilton sikli yg'li: dastur topgan yg'l” chiqariladi).

Masalan, 1-holat uchun yuqorida keltirilgan algoritmdan foydalanib, quyidagi grafning insidentlik matritsasi

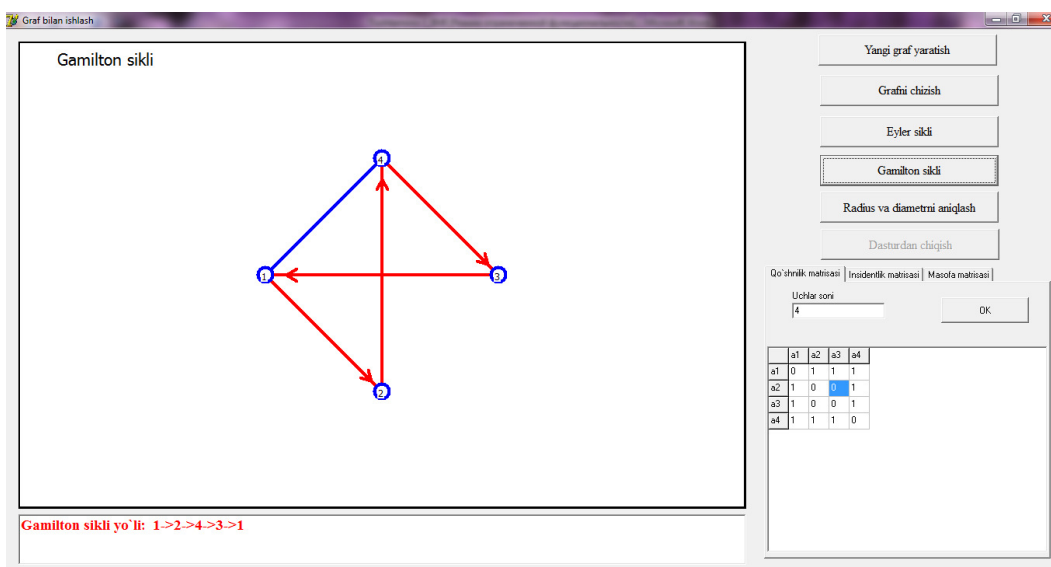
$$B = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

yordamida qurilgan grafni dasturiy vosita orqali Gamilton sikliga tekshirish natijasi quyidagicha(2.7-rasm)



2.7-rasm. Hosil qilingan grafni Gamilton sikli tekshirishda mavjud bo'lmagan holati.

2-holat uchun 2.1-masalada a) bandidagi ko'rilgan misol uchun qurilgan grafni dasturiy vosita orqali Gamilton sikliga tekshirish natijasi quyidagicha(2.8-rasm).



2.8-rasm. Hosil qilingan grafni Gamilton sikli tekshirishda mavjud bo'lgan holati.

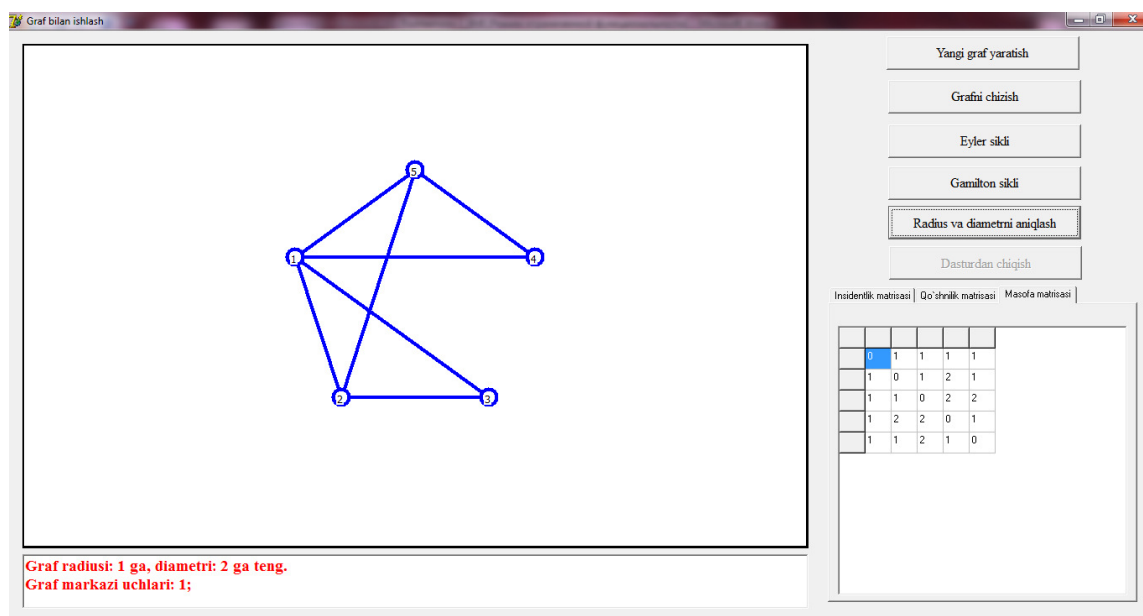
2.3-masalani a) va b) bandini yechish va undan olinadigan natijalarini hosil qilish bo'yicha ko'rsatmalar.

“Uchlar qo'shniligi” yoki “insidentlik” matritsasi yordamida hosil qilingan grafni masofa matritsasini, graf radiusi, diametri va markazini aniqlash uchun yuqoridagi ko'rsatmadagi 2.1-masalani a) yoki b) bandi dasturiy vositada bajarilgandan so'ng, “Radius va diametri aniqlash” tugmasi bosiladi.

Aniqlangan masofa matritsasini kg‘rish uchun “Masofa matritsasi” bg‘limi tanlanadi . Bu yerda masofa matritsasini jadvalda kg‘rish mumkin.

Graf radiusi,diametri va markaz uchlari xabar oynasidan kg‘rish mumkin.

Masalan, 2.1-masalada b) bandidagi kg‘rilgan misol uchun qurilgan grafni dasturiy vosita orqali grafni masofa matritsasini, graf radiusi,diametri va markazini aniqlash natijasi quyidagicha(2.9-rasm).



2.9-rasm. Hosil qilingan grafni masofa matritsasi, graf radiusi,diametri va markaz uchlari natija oynasi.

Dasturdan chiqish uchun “Dasturdan chiqish” tugmasi bosiladi va dastur g‘z ishini yakunlaydi.

Xulosa

Bitiruv malakaviy ishida muayyan ob'yektlarning graf modelini uchlar qo'shniligi va insidentlik matritsalarini yordamida hosil qilish. Hosil qilingan graf modellarini Eyler va Gamilton sikliga tekshirish, graflarning metrik xarakteristikalarini: uchlar orasidagi masofa matritsasi, graf radiusi, diametri, markazlarini aniqlash algoritmi tuzildi va vizuallashtirilgan dasturiy vosita ishlab chiqildi. Bu dasturiy vositadan foydalanib, uchlar qo'shniligi va insidentlik matritsalarini yordamida turli graflarni hosil qilish va ularni Eyler va Gamilton sikliga tekshirish, graflarning metrik xarakteristikalarini: uchlar orasidagi masofa matritsasi, graf radiusi, diametri, markazlarini aniqlash bo'yicha bir necha masalalar ko'rib chilgan (qar. 2.1-2.3-masalalar).

Ishda olingan natijalar va unda qo'llanilgan usullardan turli iqtisodiy, ijtimoiy sohalarning ko'pgina amaliy masalalarining graf modellarini hosil qilishda, ularning dasturiy vositalarini ishlab chiqishda, "Graflar nazariyasi", "Axborot xavfsizligi", "Kompyuter injineri", "Dastur injineri" va shu kabi fanlarning amaliy mashg'ulotlari o'quv jarayonlarida dasturiy vosita sifatida foydalanish mumkin.

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. То'rayev Н.Т., 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>

1-ilova. Muayyan ob'ektlarning graf modelini hosil qilish.

```
procedure TForm1.hosilNuqta(n:integer);
var x,y,i,r:integer;
    da:double;
begin
    x:=Image1.Width div 2;
    y:=Image1.Height div 2;
    r:=150;
    da:=2*pi/n;
    for i:=1 to n+1 do
        begin
            arr[i].x:=x+round(r*cos(pi/2+i*da));
            arr[i].y:=y-round(r*sin(pi/2+i*da));
        end;
    end;
procedure TForm1.StringGridtoMatrix;
var i,j,n,m:integer;
begin
    n:=StringGrid1.RowCount-1;
    m:=StringGrid1.ColCount-1;
    for i:=1 to n do
        for j:=1 to m do
            A[i,j]:=StrToInt(StringGrid1.Cells[j,i]);
        end;
    end;
procedure TForm1.chizishChiziq( i,j:integer);
var p1,p2:TPoint;
begin
    Image1.Canvas.Pen.Style:=psDot;
    Image1.Canvas.Pen.Width:=4;
    Image1.Canvas.Pen.Color:=clBlue;
    Image1.Canvas.MoveTo(arr[j].X,arr[j].Y);
    Image1.Canvas.LineTo(arr[i].X,arr[i].Y);
end;
procedure TForm1.chizishGraf;
var i,j,n:integer;
begin
    Image1.Canvas.Pen.Color:=clBlack;
    Image1.Canvas.Pen.Width:=4;
    Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
    n:=StringGrid1.RowCount-1;
    hosilNuqta(n);
    StringGridtoMatrix;
    for i:=1 to n do
        for j:=i to n do
            begin
                if(A[i,j]=1) then chizishChiziq(i,j);
            end;
        Image1.Canvas.Pen.Color:=clBlue;
        for i:=1 to n do
```

```

begin
    Image1.Canvas.Ellipse(arr[i].X-10,arr[i].Y-10,arr[i].X+10,arr[i].Y+10);
    Image1.Canvas.TextOut(arr[i].X-5,arr[i].Y-5,IntToStr(i));
end;
end;
procedure TForm1.InsedenttoQushnilik;
var i,j,i1,j1,l,m:integer;
begin
    m:=StringGrid3.RowCount-1;
    l:=StringGrid3.ColCount-1;
    StringGrid3toMatrix;
    for i:=1 to 100 do
        for j:=1 to 100 do
            A[i,j]:=0;
        for i:=1 to l do
            begin
                i1:=0; j1:=0;
                for j:=1 to m do
                    if(Ins[j,i]=1) then
                        begin
                            if(i1>0) then
                                j1:=j
                            else
                                i1:=j;
                        end;
                    if(i1+j1>1) then
                        begin
                            A[i1,j1]:=1;
                            A[j1,i1]:=1;
                        end;
                    end;
                end;
            end;
        end;
    end;
end;

```

2-ilova. Berilgan graflarni Eyler va Gamilton sikliga tekshirish.

```

procedure TForm1.EylerSikli;
const
    maxn = 100;
var
    e,was:array[1..maxn,1..maxn]of integer;
    ne:array[1..maxn]of integer;
    stack:array[1..maxn*maxn]of integer;
    j,i,n,u,v,top:longint;
    ok:boolean;
begin
    n:=StringGrid1.RowCount-1;
    // m:=0;
    StringGridtoMatrix;
    for i:=1 to n+1 do
        begin
            ne[i]:=0;
            for j:=1 to n+1 do

```

```

begin
  e[i,j]:=0;
  stack[(n+1)*(i-1)+j]:=0;
  was[i,j]:=0;
end;
end;
for i:=1 to n do
for j:=i to n do
begin
  if(A[i,j]=1) then
  begin
    u:=i;
    v:=j;
    inc(ne[u]); e[u,ne[u]]:=v;
    inc(ne[v]); e[v,ne[v]]:=u;
  end;
end;
j:=0;
for i:=1 to n do
begin
  if ne[i]=0 then inc(j);
  if ne[i] mod 2=1 then
  begin
    RichEdit1.Text:='Berilgan grafda Eyler sikli mavjud emas.';
    exit;
  end;
end;
if j=n then
begin
  RichEdit1.Text:='Berilgan graf bir-biri bilan bog'lanmagan.';
  exit;
end;
top:=1; stack[1]:=1;
j:=0;
for i:=1 to 100 do
  eys[i]:=0;
while top>0 do begin
  u:=stack[top]; ok:=true;
  for i:=1 to ne[u] do
  begin
    v:=e[u,i];
    if was[u,v]=0 then
    begin
      inc(top); stack[top]:=v;
      was[u,v]:=1;was[v,u]:=1;
      ok:=false; break;
    end;
  end;
end;
if ok then
begin
  dec(top);
  inc(j);

```

```

        eys[j]:=u;
    end;
end;
chizishGrafEylar(j);
end;
procedure TForm1.chizishGrafEylar(n:integer);
var i:integer;
begin
    RichEdit1.Text:='Eylar sikli yg'li:  ';
    for i:=2 to n do
        begin
            chizishChiziqEylar(eys[i-1],eys[i]);
            RichEdit1.Text:=RichEdit1.Text+IntToStr(eys[i-1])+'->';
        end;
    RichEdit1.Text:=RichEdit1.Text+IntToStr(eys[n]);
    for i:=1 to StringGrid1.RowCount-1 do
        begin
            Image1.Canvas.Pen.Color:=clBlue;
            Image1.Canvas.Ellipse(arr[i].X-10,arr[i].Y-10,arr[i].X+10,arr[i].Y+10);
            Image1.Canvas.TextOut(arr[i].X-5,arr[i].Y-5,IntToStr(i));
        end;
    end;
end;
procedure TForm1.print_gamilton_c(n:integer);
var
    p,i:integer;
begin
    RichEdit1.Text:='Gamilton sikli yg'li:  ';
    for p:=1 to n-1 do
        begin
            chizishChiziqEylar(path[p-1]+1,path[p]+1);
            RichEdit1.Text:=RichEdit1.Text+ IntToStr(path[p-1]+1)+'->';
        end;
        chizishChiziqEylar(path[n-1]+1,path[0]+1);
        RichEdit1.Text:=RichEdit1.Text+IntToStr(path[n-1]+1)+'-
>'+IntToStr(path[0]+1)+' ';
        for i:=1 to StringGrid1.RowCount-1 do
            begin
                Image1.Canvas.Pen.Color:=clBlue;
                Image1.Canvas.Ellipse(arr[i].X-10,arr[i].Y-10,arr[i].X+10,arr[i].Y+10);
                Image1.Canvas.TextOut(arr[i].X-5,arr[i].Y-5,IntToStr(i));
            end;

        end;
end;
function TForm1.gamilton(k:integer):integer;
var
    v,gl,n:integer;
begin
    n:=StringGrid1.RowCount-1;
    gl:=0;
    v:=-1;
    while ((v<n) and (gl=0)) do begin
        inc(v);

```

```

    if (a[v+1,path[k-1]+1]>0) or (a[path[k-1]+1,v+1]>0) then begin
      if (k=n) and (v=v0) then gl:=1
      else if (c[v]=-1) then begin
        c[v]:=k;
        path[k]:=v;
        gl:=gamilton(k+1);
        if (gl=0) then c[v]:=-1;
        end else continue;
      end;
    end;
  result:=gl;
end;
procedure TForm1.Button4Click(Sender: TObject);
var j,n:integer;
begin
  Button2Click(Self);
  n:=StringGrid1.RowCount-1;
  for j:=0 to n-1 do c[j]:=-1;
  path[0]:=v0;
  c[v0]:=v0;
  if (gamilton(1)>0) then print_gamilton_c(n) else
    RichEdit1.Text:='Gamilton sikli berilgan grafda mavjud emas.';
end;

```

3-ilova. Graflarning metric xarakteristikalarini aniqlash.

```

procedure TForm1.matritsaD;
var i,j,n,m:integer;
begin
  n:=StringGrid1.RowCount-1;
  for i:=1 to n do
    for j:=1 to n do
      begin
        D[i,j]:=A[i,j];
        if(D[i,j]=0) then
          D[i,j]:=100000;
        end;
      for i:=1 to n do
        D[i,i]:=0;
      for m:=1 to n do
        for i:=1 to n do
          for j:=1 to n do
            D[i,j]:=min(D[i,j],D[i,m]+D[m,j]);
            StringGrid2.RowCount:=n+1;
            StringGrid2.ColCount:=n+1;
          for i:=1 to n do
            for j:=1 to n do
              begin
                if(D[i,j]=100000) then
                  StringGrid2.Cells[i,j]:='infity'
                else
                  StringGrid2.Cells[i,j]:=IntToStr(D[i,j]);
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```



```

end;
procedure TForm1.maxDElement;
var i,n:integer;
begin
n:=StringGrid1.RowCount-1;
for i:=1 to n do
W[i]:=MaxIntValue(D[i]);
end;
function TForm1.GrafRadius:integer;
var i,n,min1:integer;
begin
n:=StringGrid1.RowCount-1;
min1:=W[1];
for i:=2 to n do
min1:=min(min1,W[i]);
result:=min1;
end;
function TForm1.GrafDiametr:integer;
var i,n,max1:integer;
begin
n:=StringGrid1.RowCount-1;
max1:=W[1];
for i:=2 to n do
max1:=max(max1,W[i]);
result:=max1
end;
procedure TForm1.BitBtn2Click(Sender: TObject);
var i,n,gr:integer;
begin
Button2Click(Self);
matritsaD;
maxDElement;
gr:=GrafRadius;
if(gr=100000) then
RichEdit1.Text:='Graf bog'lamli emas'
else
begin
RichEdit1.Text:='Graf radiusi: '+IntToStr(GrafRadius)+' ga';
RichEdit1.Text:=RichEdit1.Text+', diametri: '+IntToStr(GrafDiametr)+' ga
teng.';

n:=StringGrid1.RowCount-1;
RichEdit1.Text:=RichEdit1.Text+#13+'Graf markazi uchlari: ';
for i:=1 to n do
if(W[i]=gr) then
RichEdit1.Text:=RichEdit1.Text+IntToStr(i)+' ';
end;
end;
end;

```