МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО

МЕТОДИЧНІ ВКАЗІВКИ
ЩОДО ВИКОНАННЯ РОЗРАХУНКОВО - ГРАФІЧНОЇ РОБОТИ
З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«**ДИСКРЕТНА МАТЕМАТИКА**»
ДЛЯ СТУДЕНТІВ ДЕННОЇ ФОРМИ НАВЧАННЯ ЗІ СПЕЦІАЛЬНОСТІ
123 – «КОМП’ЮТЕРНА ІНЖЕНЕРІЯ»

КРЕМЕНЧУК 2019

Методичні вказівки щодо виконання розрахунково-графічної роботи з навчальної дисципліни «Дискретна математика» зі спеціальності 123 – «Комп’ютерна інженерія»

Укладачі: старш. викл. В. Ю. Бельська,

старш. викл. А. Л. Юдіна

Рецензент к. т. н., доц. В. М. Сидоренко

Кафедра комп’ютерних та інформаційних систем

Затверджено методичною радою КрНУ імені Михайла Остроградського

Протокол № від __ _______ 2019 р.

Голова методичної ради _________________ проф. В. В. Костін

ЗМІСТ

ВСТУП

Дисципліна «Дискретна математика» є базовою для підготовки бакалаврів зі спеціальності 123 – «Комп'ютерна інженерія». Отримані знання будуть використані студентами при вивченні дисциплін «Алгоритми та методи обчислень», «Прикладне програмування», «Програмування», «Архітектура ЕОМ2, «Алгоритми та методи обчислень», «Комп'ютерні мережі» та ін.

Розрахунково-графічна робота з дисциплін «Дискретна математика» охоплює основні розділи: теорія множин, задачі оптимізації на графах, комбінаторика. До кожного завдання подано основні теоретичні відомості, які студенти безпосередньо використовують для самостійного виконання семестрових завдань. Наведені приклади містять повну інформацію про основні етапи алгоритмів розв'язування запропонованих до виконання завдань роботи.

Досить вдалим доповненням цих методичних вказівок є методичні вказівки щодо виконання практичних та самостійної робіт студентів з дисципліни «Дискретна математика», частина 1: «Множини, пошукові задачі на графах».

У результаті виконання розрахунково-графічної роботи студенти повинні:

– *знати*: математичний апарат дискретної математики, основи математичної логіки, алгоритми та засоби пошуку оптимальних розв'язань типових задач;

– *уміти*: знаходити найбільш ефективний для розв'язання конкретної задачі математичний апарат, моделювати та досліджувати елементарні логічні та дискретні схеми.

1 ТЕОРІЯ МНОЖИН. ОСНОВНІ ОПЕРАЦІЇ НАД МНОЖИНАМИ

Множиною називається сукупність визначених цілком помітних об'єктів, розглянутих як єдине ціле, тобто сукупність, сім'я, комплекс довільних об'єктів, що цілком розрізняються нашою думкою або інтуїцією. Окремі об'єкти, з яких складається множина, називаються *елементами* множини.

Об'єднанням множин X і Y називається множина, що складається зі всіх тих і тільки тих елементів, що належать хоча б одній з множин X, Y, тобто належать X або належать Y. Позначається через $X \cup Y$.

$$x \in X \cup Y \leftrightarrow x \in X \text{ або } x \in Y.$$

Наприклад: якщо $X = \{1,2,3,4,5\}$ і $Y = \{2,4,6,7\}$, то $X \cup Y = \{1,2,3,4,5,6,7\}$.

Перетином множин X і Y називається множина, що складається зі всіх тих і тільки тих елементів, що належать як множині X, так і множині Y, позначається $X \cap Y$ (іноді називається добутком множин):

$$x \in X \cap Y \leftrightarrow x \in X \text{ і } x \in Y.$$

Наприклад: якщо $X = \{1,2,3,4,5\}$ і $Y = \{2,4,6,7\}$, то $X \cap Y = \{2,4\}$.

Множини X і Y називаються множинами, що не *перетинаються*, якщо вони не мають загальних елементів, тобто якщо $X \cap Y = \varnothing$.

Множини X і Y знаходяться в загальному положенні, якщо виконуються три умови:

1) $\exists\, x \in X$ та $x \notin Y$;

2) $\exists\, y \in Y$ та $y \notin X$;

3) $\exists\, x \in X$ та $x \in Y$.

Між двома множинами X і Y може бути одне з п'ятьох відношень: $X = Y$; $X \subset Y$; $Y \subset X$; $X \cap Y = \varnothing$; X та Y знаходяться в загальному положенні.

Операції об'єднання і перетину множин володіють комутативним і асоціативним властивостями:

$$X \cap Y = Y \cap X;$$

$$(X \cap Y) \cap Z = X \cap (Y \cap Z) = X \cap Y \cap Z.$$

Різницею множин X і Y називається множина, що складається зі всіх тих і тільки тих елементів, що належать множині X і не належать множині Y, позначається X\Y.

$$x \in X \setminus Y \leftrightarrow x \in X \text{ і } x \notin Y.$$

Наприклад: якщо X ={1,2,3,4,5} і Y={2,4,6,7}, то X\Y={1,3,5}.

На відміну від двох попередніх операцій, різниця двох множин визначена для двох множин, тобто тільки двомісна, некомутативна ($X \setminus Y \neq Y \setminus X$).

Роль одиниці в алґебрі множин має "універсальна множина" U, що задовольняє умову: $X \cap U = X$, $X \cup U = U$ (немає аналогії у звичайній алґебрі) .

Доповненням множини X називається множина всіх елементів, що не належать X, а належать $\overline{X} = U \setminus X$.

$$\overline{X} = \{x : x \in U \ i \ x \notin X\}.$$

З визначення випливають властивості: $X \cap \overline{X} = \varnothing$; $X \cup \overline{X} = U$.

Справедливі тотожністі алґебри множин:

1) $(X \cup Y) \cap Z = (X \cap Z) \cup (Y \cap Z)$;

2) у звичайній алґебрі ми не можемо замінити в дистрибутивному законі дію додавання множенням, а дію множення додаванням, оскільки це призводить до хибного виразу (ab)+c = (a+c)(b+c). Але в алґебрі множин подібний вираз має зміст: $(X \cap Y) \cup Z = (X \cup Z) \cap (Y \cup Z)$;

3) якщо $Y \subseteq X$, то $X \cap Y = Y$, $X \cup Y = X$;

4) якщо Y = X, то $X \cap X = X$, $X \cup X = X$;

5) $\overline{X \cup Y} = \overline{X} \cap \overline{Y}$;

6) $\overline{X \cap Y} = \overline{X} \cup \overline{Y}$.

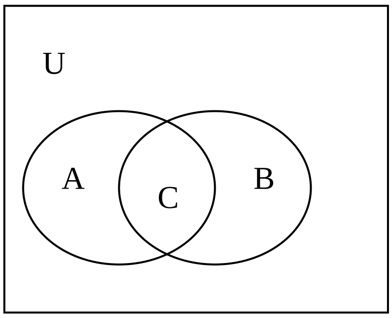
Тотожності 5) і 6) називаються тотожностями Де-Моргана.

Розглянемо *приклад 1.1.* Задано універсальну множину U={1,2,3,…,60} та три її підмножини $A=\{a \in U: a=2 \cdot i\}$, $B=\{b \in U: b=3 \cdot i\}$, $C=\{c \in U: c=6 \cdot i\}$, i=1,2,…

Необхідно зобразити за допомогою діаграм Ейлера – Венна відношення між множинами U, A, B, C. Зазначити кількість елементів множин, обумовлених формулою алгебри множин $F = \left(\overline{\overline{A}\cup\overline{B}}\right)\cup\left(\overline{\overline{B}\cup\overline{C}}\right)\cap\left(\overline{A\cap\overline{B}\cap C}\right)$.

Розв'язання

Універсальну множину зручно зображувати графічно у вигляді множини точок прямокутника. Окремі області у середині цього прямокутника будуть означати різні підмножини універсальної множини. Зображення множин у прямокутнику, що являє собою універсальна множина, називається діаграмою Ейлера – Венна. Виконаємо аналіз можливого зв'язку між елементами заданих множин A, B, C. Множини A і B знаходяться в загальному відношенні. Елементи множини C належать як множині A (число 6 кратне 2), так і множині B (число 6 кратне 3), що збігається з правилом перетину двох множин, тобто множина C = A∩B.



Для визначення кількості елементів множин, обумовлених формулою F, виконаємо спрощення за допомогою тотожностей алґебри множин.

$\overline{\overline{A}\cup\overline{B}} = A\cap B = C$; - застосовано закон Де-Моргана

$\overline{\overline{B}\cup\overline{C}} = B\cap C = C$;- застосовано закон Де-Моргана

$\overline{A\cap\overline{B}\cap C} = (\textit{тотож.}\ 3) = \overline{C\cap\overline{B}} = \overline{C\setminus B} = \overline{\varnothing} = U$;

$$F = \left(\overline{\overline{A}\cup\overline{B}}\right)\cup\left(\overline{\overline{B}\cup\overline{C}}\right)\cap\left(\overline{A\cap\overline{B}\cap C}\right) = C\cup C\cap U = C\cap U = C.$$

У результаті отримали: формула F алґебри множин описує множина C, яка складається з 10 елементів, а саме {6, 12, 18,…, 60}

Завдання № 1

Задано універсальну множину U={1,2,3,…,60} та чотири її підмножини A={a∈U: a=p·i}, B={b∈U: b=q·i}, C={c∈U: c=r·i}, D={d∈U: d=s·i}, i=1,2,… Значення p, q, r, s наведено в таблиці.

Пара-метр	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
p	5	3	1	4	2	4	3	2	5	4	1	2	1	3	5
q	3	6	2	2	4	8	2	6	3	6	4	4	3	5	4
r	6	4	3	6	3	6	6	12	10	2	3	6	2	2	6
s	2	9	4	3	6	3	5	8	9	3	8	5	10	4	2
Пара-метр	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
p	1	6	8	2	10	2	4	3	3	3	2	3	1	2	10
q	4	3	4	5	5	3	5	2	5	6	4	5	5	10	2
r	2	1	1	6	1	1	1	4	6	12	24	7	6	3	3
s	6	7	3	7	6	5	2	8	1	5	6	21	15	4	9

Зобразити за допомогою діаграм Ейлера – Венна відношення між множинами U, A, B, C, D.

Зазначити кількість елементів множин, обумовлених формулами Fi алгебри множин (i – номер варіанта).

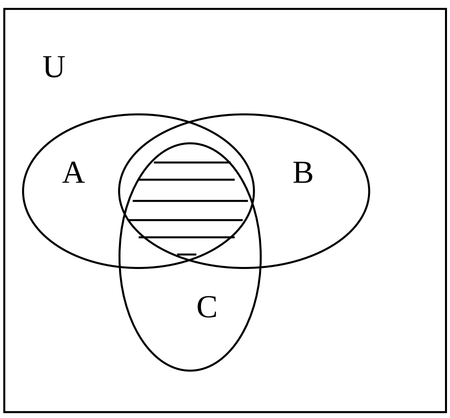
$F_1 = \overline{\overline{A}\cup\overline{B}}\cap\overline{C\cap D}\cup(\overline{C}\cap\overline{D})$	$F_{16} = \left(\overline{\overline{A}\cup B\cup\overline{C}}\right)\cap\left(\overline{A\cup\overline{B}\cup C\cup\overline{D}}\right)$
$F_2 = A\cap\left(\overline{A\cap\overline{C\cup D}\cup B}\right)$	$F_{17} = \left(\overline{\overline{A}\cup\overline{B}}\right)\cup\left(\overline{\overline{B}\cup\overline{C}}\right)\cap\left(\overline{A\cap\overline{B}\cap C\cap D}\right)$
$F_3 = \overline{B\cap C}\cup\overline{\overline{A}\cap D}\cup\left(A\cap B\cap\overline{C}\right)$	$F_{18} = \left(\overline{(A\cup B)\cap C}\right)\cup\left(\overline{\overline{C}\cup\overline{D}}\right)$
$F_4 = \overline{B}\cap\left(\overline{A\cup C}\cup\overline{C\cap D}\right)$	$F_{19} = \left(\overline{A\cup\overline{B}\cup C\cup\overline{D}}\right)\cup\left(\overline{\overline{A}\cap B\cap\overline{C}\cap D}\right)$
$F_5 = (A\cap B)\cup\left(\overline{A}\cap C\cap D\right)\cup\left(A\cap\overline{B}\right)$	$F_{20} = \left(A\cap\overline{D}\right)\cap\left(\overline{A\cap B\cap D}\right)\cap(A\cap B)$
$F_6 = \left(A\cap\overline{C}\cap\overline{D}\right)\cup\left(\overline{A\cap\overline{B}}\right)\cup(A\cap B)$	$F_{21} = (A\cup C\cap D)\cap\left(\overline{\overline{B}\cap\overline{D}}\right)\cup D$
$F_7 = \left(\overline{A}\cup B\cup\overline{C}\right)\cap\left(\overline{A\cup\overline{B}\cup C\cup\overline{D}}\right)$	$F_{22} = \overline{(A\cup B)\cap\left(\overline{A}\cup C\cup D\right)}\cap\left(\overline{B}\cap\overline{D}\right)$
$F_8 = \left(\overline{A\cup\overline{B}\cup C}\right)\cap\left(\overline{\overline{A}\cap B\cap\overline{C}\cap D}\right)$	$F_{23} = \left(\overline{\overline{D}\cap\overline{B\cup C}}\right)\cap\left(\overline{C\cap D\cup\overline{A}}\right)$

$F_9=\left(\overline{A\cup B}\right)\cap\left(\overline{B\cup C}\right)\cap\left(\overline{C\cup D}\right)\cap\left(\overline{A\cap\overline{B}}\right)$	$F_{24}=\left(\overline{\overline{D}\cup\overline{C}}\right)\cap\left(\overline{D\cap C}\right)\cup\left(\overline{A\cap D\cap\overline{C}}\right)$
$F_{10}=\left(A\cap\overline{B}\cap\overline{C}\right)\cup\left(\overline{\overline{A}\cap C\cap D}\right)$	$F_{25}=\overline{B}\cup\left(\overline{A\cup C}\cap\overline{C\cap D}\right)$
$F_{11}=\left(A\cup C\cap D\right)\cup\left(\overline{B\cap D}\right)\cap D$	$F_{26}=\left(\overline{A\cup\overline{B}\cup C\cup\overline{D}}\right)\cap\left(\overline{\overline{A}\cap B\cap\overline{C}}\right)$
$F_{12}=\left(\overline{A\cap B\cup C}\right)\cap\left(\overline{C\cap D\cup\overline{A}}\right)$	$F_{27}=\left(A\cap C\right)\cup\left(A\cap D\right)\cap\left(\overline{B\cap C\cup\overline{D}}\right)$
$F_{13}=\left(\overline{B\cap C}\right)\cap\left(D\cup\overline{C}\right)\cup\left(\overline{A\cap D\cap\overline{C}}\right)$	$F_{28}=\left(A\cap B\right)\cup\left(\overline{\overline{A}\cap D}\right)\cup\left(A\cap\overline{B}\cap\overline{D}\right)$
$F_{14}=\left(\overline{\overline{C}\cup\overline{D}\cup B}\right)\cap\left(A\cap B\cup\overline{C}\right)$	$F_{29}=\left(\overline{A\cup B}\right)\cap\left(\overline{B\cup C}\right)\cap\left(\overline{C\cup D}\right)\cap\left(\overline{A\cap\overline{B}}\right)$
$F_{15}=\left(A\cup B\right)\cap\left(\overline{A\cap\overline{C\cup D}}\right)$	$F_{30}=\left(A\cup\overline{C\cap D}\right)\cup\left(\overline{B\cap D}\right)\cap D\cup\overline{C}$

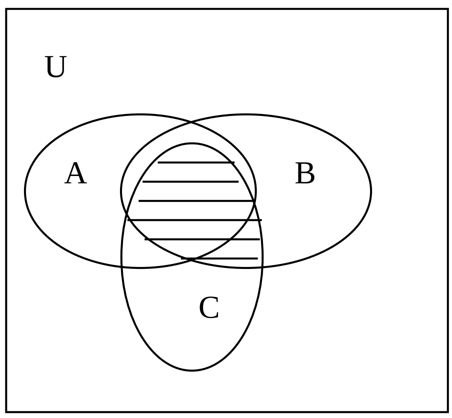
Приклад 1. 2 Задано множини A, B, C , що знаходяться в загальному положенні. Необхідно за допомогою діаграм Ейлера – Венна знайти розв'язання системи рівнянь $\begin{cases} C=A\cap B \\ B\cap U=C \\ \left(A\cap B\right)\cup\left(\overline{\overline{B}\cup\overline{C}}\right)=U \end{cases}$.

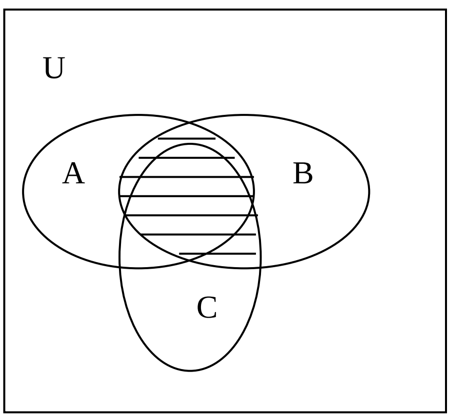
Розв`язання. Для кожного з рівнянь системи побудуємо діаграму Ейлера – Венна

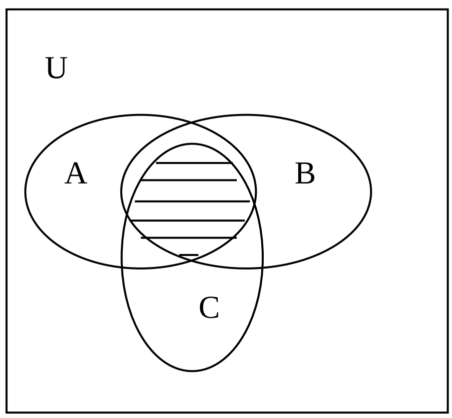
1) $C=A\cap B,\ \textit{тобто}\ C\cap A\cap B$;



2) $B\cap U=C,\ \textit{тобто}\ B\cap C$;







3) $(A\cap B)\cup\left(\overline{\overline{B}\cup\overline{C}}\right)=U$ *або* $(A\cap B)\cup\left(\overline{\overline{B}\cup\overline{C}}\right)$ *або* $(A\cap B)\cup(B\cap C)=B\cap(A\cup C)$;

4) результуюча діаграма є перетином результатів на діаграмах 1), 2), 3) – розв'язання системи збігається із зображенням на діаграмі 1).

Завдання № 2

Задано множини A, B, C, D , що знаходяться в загальному положенні. Необхідно за допомогою діаграм Ейлера – Венна знайти розв'язання системи рівнянь.

1. $\begin{cases} A=B\cap\overline{C} \\ \overline{C}\cap\overline{D}=U \\ A\cup C=B\cap D \\ C=\overline{A\cup B}\cap\overline{D} \end{cases}$

2. $\begin{cases} C=D\cup A\cup B \\ A\cap\overline{B}=\varnothing \\ \overline{A\cup B}\cup\overline{A\cup C}=D \\ A\cup B=B\cup C\cup D \end{cases}$

3. $\begin{cases} B=(A\cup D)\setminus C \\ \overline{C}\cup\overline{D}=\varnothing \\ D=A\cap B \\ \overline{\overline{A}\cap\overline{B}}\cup C=D \end{cases}$

4. $\begin{cases} B=C\cup D \\ A\cap B\cap C=\varnothing \\ (A\cap B)\cup(B\cap C)=U \\ A\setminus B=C\setminus D \end{cases}$

5. $\begin{cases} D=C\cap A\cup B \\ B\cup C=D\cap U \\ (A\cup B)\cap\left(B\cup\overline{C}\right)=D \\ \varnothing=\overline{B\cup C} \end{cases}$

6. $\begin{cases} B\cap\overline{A\cup C}=\varnothing \\ C=B\setminus D \\ A\cup C=B\setminus D \\ C\cup D\cup A=B \end{cases}$

7. $\begin{cases} A\cap B=\overline{C} \\ \overline{\overline{C}\cap\overline{D}}\cap A=U \\ (A\cup D)\setminus C=B \\ B\cup C=A\cup D\cup\overline{\varnothing} \end{cases}$

8. $\begin{cases} C\cap D=A\cup B \\ \overline{B}\cup\overline{A}=U \\ A\cap C=B\cup D \\ B=\overline{A\cup C}\cap D \end{cases}$

9. $\begin{cases} D=(A\cap B)\cup\overline{C} \\ \overline{A}\cap\overline{D}=\varnothing \\ C=A\cup B \\ \overline{\overline{A}\cup\overline{D}}\cap B=C \end{cases}$

10. $\begin{cases} A=C\cap D \\ A\cup B\cap C=\varnothing \\ \left(\overline{A}\cup\overline{C}\right)\cup\left(\overline{A}\cap C\right)=D \\ A\cup B=C\cup\overline{D} \end{cases}$

11. $\begin{cases} C=D\cup A\cap B \\ B\cap C=D\cup\varnothing \\ (A\cap B)\cup(B\cap C)=\varnothing \\ U=B\cap C \end{cases}$

12. $\begin{cases} B\cup\overline{A\cap\overline{C}}=U \\ \overline{C}=B\setminus\overline{D} \\ A\cap C=B\cup D \\ C\cap D\cap A=\overline{B} \end{cases}$

13. $\begin{cases} \overline{\overline{A}\cup\overline{\overline{B}}}=C \\ \overline{\overline{C}\cup\overline{A}}\cup D=\varnothing \\ (A\cap D)\cup\overline{C}=\overline{B} \\ B\cap C=A\cup D \end{cases}$

14. $\begin{cases} \overline{\overline{C}\cup D}=A \\ B\cap\overline{A}=\varnothing \\ A\cup C=B\cap D \\ \overline{B}=A\cup D \end{cases}$

15. $\begin{cases} A=D\cup B\cap\overline{C} \\ A\cup D=\overline{U} \\ \overline{\overline{A}\cup\overline{B}}\cup C\cap\overline{B}=U \\ D=B\cap C\cup\overline{A} \end{cases}$

16. $\begin{cases} C\cup\overline{\overline{B}\cap\overline{C}}=D \\ \overline{C}=B\cup D \\ A\cup D=U\cap\overline{B} \\ C\cup D\cap A=\varnothing \end{cases}$

17. $\begin{cases} A\cup B=C\cap\varnothing \\ \varnothing=\overline{C}\cup\overline{D}\cup A \\ A\cap D=U \\ (A\cap\overline{B})\cup(A\cap B)=C\cup D \end{cases}$

18. $\begin{cases} C\cap\overline{B\cup\overline{C}}=U \\ C=B\cup D\cup A \\ A\setminus D=B\cup C \\ C\cap D\cup\overline{A}=U \end{cases}$

19. $\begin{cases} C=A\cap\overline{D}\cup B \\ C\cup\overline{D}=B \\ D\cap A\cap B=U \\ A\cup B\cap\overline{C}=\overline{D} \end{cases}$

20. $\begin{cases} C\cap D\cap A=A\cup B \\ A\cup B=\varnothing\cup C \\ A\cup B\cup C=D\cap U \\ A=C\cup B\cap\overline{C} \end{cases}$

21. $\begin{cases} A=D\cap C\cup A \\ B\cap C=\overline{D\cup\overline{\varnothing}} \\ A\cap\overline{B}\cup\overline{C\cup\overline{A}}=U \\ D=A\cap B \end{cases}$

22. $\begin{cases} D=B\cup C \\ C=U\cup\varnothing \\ A\cap C=B\setminus D \\ A\cap C=(A\cup D)\setminus C \end{cases}$

23. $\begin{cases} A=B\cup C \\ \overline{C}\cap\overline{D}=C\cup A \\ \varnothing=A\cup B\cap\overline{\overline{A}\cap D} \\ A\cup D=B\cup C\cup\varnothing \end{cases}$

24. $\begin{cases} B=C\cap D \\ C\cap D\cup A=B \\ \overline{C}=C\cap\overline{B} \\ B\cap D\cup A=\varnothing \end{cases}$

25. $\begin{cases} C=A\cup B\cup D \\ B\cap\overline{D}\cup C=\varnothing \\ A\cap B\cup C\cap U=A\cap D \\ \overline{A\cup\overline{B}}=B\setminus C\cup D \end{cases}$

26. $\begin{cases} D=\overline{C\cap\overline{B}} \\ C\cap\overline{D}=U \\ B\cup D\cap A=A\cup B \\ A\cap\overline{C}=D\cup A \end{cases}$

27. $\begin{cases} A\cup B=D\setminus C \\ A\cap B\cup C=\overline{U\cup\overline{\varnothing}} \\ C\cup D\cup A=B \\ B\cap C=C\cap\overline{D} \end{cases}$

28. $\begin{cases} B\cup A=C\cap D \\ A\cup B\cup C=U \\ A\cap\overline{B}\cup C=C\cap\overline{D} \\ B\cap C=A\cup\overline{D}\cup\overline{B} \end{cases}$

29. $\begin{cases} C\cap D=\overline{C\cap\overline{B}} \\ B\cup D=\varnothing \\ \overline{A\cap\overline{B}}\cup D=B\cap C \\ C=A\cup B\cup\overline{\varnothing} \end{cases}$

30. $\begin{cases} A\cap C=C\cap D \\ B=D\cap\overline{C} \\ A\cup\overline{B}\cap\overline{C}=A\cup D \\ D=A\cup\overline{B\cap\overline{C}}\cup\overline{D} \end{cases}$

2 ЗАДАЧІ ОПТИМІЗАЦІЇ НА ГРАФАХ

Графом назвемо деяку множину точок площини X, що називаються вершинами, і множину спрямованих відрізків U, що з’єднують усі або деякі з вершин, названих дугами. Математичне визначення графа:

1) граф G – це пара множин X и U, G = (X,U);

2) граф G – це пара множин (X, Г), що складаються із множини вершин X й відображення Г, заданого на цій множині: G=(X, Г). Це визначення збігається з визначенням відношення на множині.

Дуга – спрямований відрізок, що з’єднує будь-які дві вершини графа. Позначається: $u = (a, b)$ або $u_{a,b}$, якщо дуга виходить із вершини а і входить у вершину b. Дві вершини a і b називаються суміжними, якщо вони різні й з’єднані дугою, що йде з вершини a у вершину b. Дуга u називається інцидентною вершині a, якщо вона заходить у цю вершину або виходить із неї. Якщо кожна вершина графа G(X, Г) з'єднана дугою, то граф називається орієнтованим. Якщо розглядається граф без обліку орієнтації, то він називається неорієнтованим.

Шляхом у графі G називається послідовність дуг $\mu = (u_1, u_2, \ldots, u_k)$, у якій кінцева вершина кожної попередньої дуги збігається з початковою вершиною наступної. Шлях можна також задати за допомогою переліку вершин, що складають послідовність дуг, без зміни порядку їх проходження в дугах. У такому разі довжиною шляху $\mu = (u_1, u_2, \ldots, u_k)$ називається кінцеве або нескінченне число $\ell(\mu) = k$, яке дорівнює або числу дуг, або сумі довжин дуг, що складають шлях μ.

Для неорієнтованого графа поняття дуга, шлях, контур заміняється відповідно на ребро, ланцюг, цикл.

З поняттям неорієнтованого графа пов’язана важлива характеристика, що називається зв’язністю графа. Граф G = (X,U) називається зв’язним, якщо будь-які дві його вершини можна з’єднати ланцюгом. Якщо граф G не зв’язний, то його можна розбити на такі підграфи G_i, що всі вершини в кожному підграфі зв'язні, а вершини з різних підграфів незв'язні. Такі підграфи називаються компонентами зв'язності. Оскільки в орієнтованому графі враховується орієнтація дуг, то існує поняття надто зв'язного графа. Орієнтований граф називається сильно зв'язним, якщо для будь-яких двох вершин х і у ($x \neq y$) існує шлях, що йде з х в у.

Окремим випадком неорієнтованого графа є дерево. Дерево – кінцевий зв'язний граф, що не має циклів і складається не менше ніж із двох вершин. Основні властивості:

– дерево не містить циклів, але додавання ребра між будь-якими двома несуміжними вершинами призводить до появи одного елементарного циклу;

– дерево – зв'язний граф, але втрачає цю характеристику після видалення будь-якого ребра;

– довільна пара вершин, з'єднана ланцюгом, і тільки одним;

– дерево, побудоване на n вершинах, має n-1 ребро.

Матриця R = || rij || порядку n×n називається матрицею суміжностей, де

$$r_{ij} = \begin{cases} 1, & \textit{якщо вершина } i \textit{ суміжна з вершиною } j; \\ 0, & \textit{в іншому випадку}. \end{cases}$$

Якщо кожній дузі приписати вагу, то можна скласти матрицю ваги C = || cij || порядку n×n, де $c_{ij} = \begin{cases} \textit{вага дуги}, & \textit{якщо вершина } i \textit{ суміжна з вершиною } j; \\ \infty \quad , & \textit{в іншому випадку}. \end{cases}$

Матриця S = || sij || порядку n×m називається матрицею інциденцій, де

$$s_{ij} = \begin{cases} 1, & \textit{якщо дуга } u_{ij} \textit{ виходить з } x_i; \\ -1, & \textit{якщо дуга } u_{ij} \textit{ заходить в } x_i; \\ 0, & \textit{в іншому випадку}. \end{cases}$$

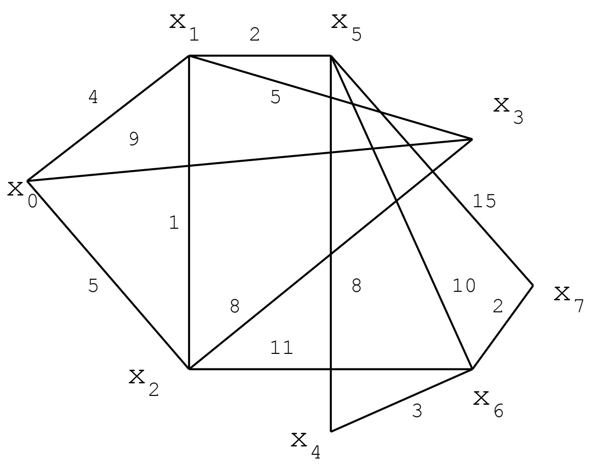
Слід зазначити, що у вигляді матриці інциденцій можна зобразити тільки орієнтований граф без петель.

Цикломатичне число. Нехай G – неорієнтований граф, який має n вершин, m ребер та r компонент зв'язності. Цикломатичним числом графа G називається число $\nu(G) = m - n + r$. Цикломатичне число не може бути неґативним і дорівнює нулю тільки в тому випадку, якщо граф не має циклів. Фізичний зміст: $\nu(G)$ дорівнює найбільшому числу незалежних циклів графа.

Хроматичне число. Граф G називається p-хроматичним, якщо його вершини можна замалювати p різними кольорами так, щоб ніякі дві суміжні вершини графа не було зарисовано одним кольором. Найменше число p, за яким граф є p-хроматичним, називається хроматичним числом і позначається $\gamma(G)$.

2.1 Пошук найкоротшого шляху між двома вершинами графа індексним методом

Серед індексних методів розглянемо метод Форда та його інтерпретацію для машинної реалізації – алгоритм Дейкстри.



Приклад 2.1 Граф G поданий графічно. Необхідно, використовуючи метод Форда, знайти найкоротшу відстань від вершини x_0 до вершини x_7 і довжину цього ланцюга.

Розв'язання. V0=0, V1=…=V_7=∞...

Розглядаємо всі вершини, суміжні з фіксованою, і вибираємо для розгляду індексу ту, довжина ребра якої має меншу вагу. Це вершина x1.

$$V_1 = min\begin{Bmatrix} \infty \\ V_0 + l_{01} = 0 + 4 = 4 \end{Bmatrix} = 4, \quad \mu_1 = \{x_0, x_1\}.$$

Далі варто розглянути вершину x_2.

$$V_2 = min\begin{Bmatrix} \infty \\ V_0 + l_{02} = 0+5=5 \\ V_1 + l_{12} = 4+1=5 \end{Bmatrix} = 5, \quad \mu_2^1 = \{x_0, x_2\}, \quad \mu_2^2 = \{x_0, x_1, x_2\}.$$

$$V_3 = min\begin{Bmatrix} \infty \\ V_0 + l_{03} = 0+9=9 \\ V_1 + l_{13} = 4+5=9 \\ V_2 + l_{23} = 5+8=13 \end{Bmatrix} = 9, \quad \mu_3^1 = \{x_0, x_3\}, \quad \mu_3^2 = \{x_0, x_1, x_3\}.$$

$$V_5 = min\begin{Bmatrix} \infty \\ V_1 + l_{15} = 4+2=6 \end{Bmatrix} = 6, \quad \mu_5 = \{x_0, x_1, x_5\}.$$

$$V_6 = min\begin{Bmatrix} \infty \\ V_2 + l_{25} = 5+11=16 \\ V_5 + l_{56} = 6+10=16 \end{Bmatrix} = 16, \quad \begin{matrix} \mu_6^1 = \{x_0, x_2, x_6\} \\ \mu_6^2 = \{x_0, x_1, x_2, x_6\}. \\ \mu_6^3 = \{x_0, x_1, x_5, x_6\} \end{matrix}$$

$$V_4 = min\begin{Bmatrix} \infty \\ V_5 + l_{54} = 6+8=14 \\ V_6 + l_{64} = 16+3=18 \end{Bmatrix} = 14, \quad \mu_4 = \{x_0, x_1, x_5, x_4\}.$$

$$V_7 = min\begin{Bmatrix} \infty \\ V_5 + l_{57} = 6+15=21 \\ V_6 + l_{67} = 16+2=18 \end{Bmatrix} = 18, \quad \begin{matrix} \mu_7^1 = \{x_0, x_2, x_6, x_7\} \\ \mu_7^2 = \{x_0, x_1, x_2, x_6, x_7\} \\ \mu_7^3 = \{x_0, x_1, x_5, x_6, x_7\} \end{matrix}$$

У результаті одержали, що з вершини x_0 у вершину x_7 існують три найкоротших ланцюги, довжина яких дорівнює 18.

Приклад 2.2 Знайдемо розв'язання попереднього прикладу за допомогою алгоритму Дейкстри.

Розв'язання. Запишемо вихідні дані в матричному вигляді. Побудуємо матрицю суміжності ваг $Д = \begin{pmatrix} \infty & 4 & 5 & 9 & \infty & \infty & \infty & \infty \\ 4 & \infty & 1 & 5 & \infty & 2 & \infty & \infty \\ 5 & 1 & \infty & 8 & \infty & \infty & 11 & \infty \\ 9 & 5 & 8 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 8 & 3 & \infty \\ \infty & 2 & \infty & \infty & 8 & \infty & 10 & 15 \\ \infty & \infty & 11 & \infty & 3 & 10 & \infty & 2 \\ \infty & \infty & \infty & \infty & \infty & 15 & 2 & \infty \end{pmatrix}$. Кожний крок відповідає побудові трьох векторів розмірності n, де n – кількість вершин графа:

A – логічний вектор, елемент a_i=1, якщо вершина x_i розглянута (на початку це вершина відправлення x_0) і a_i=0, якщо не розглянута;

B – поточне значення індексів вершин графа, тобто довжина найкоротших шляхів між вершинами x_0 та x_i (на початку елементи вектора B відповідають нульовому рядку матриці Д, оскільки вершина x_0 є початковою, за винятком елемента $в_0$, який дорівнює 0);

C – містить номери попередніх вершин, тобто c_i – номер вершини, яка стоїть перед і-ю у найкоротшому шляху в і-ту вершину (на початку всі елементи c_i=0, оскільки початкова вершина x_0 і її номер дорівнює 0).

i	0	1	2	3	4	5	6	7
A_i	1	0	0	0	0	0	0	0
B_i	0	4	5	9	∞	∞	∞	∞
C_i	0	0	0	0	0	0	0	0

Серед елементів $в_i$ вибираємо найменший для нерозглянутих вершин (яким відповідає a_i=0). Це елемент $в_1$=4.

Отож, вершина 1 стає поточною і головною для виконання всіх перетворень цього кроку.

Виконуємо такі перетворення:

a_1=1;

для кожного елемента $в_i$, який відповідає a_i=0, перевіряємо наявність більш короткого шляху в і-ту вершину з 0-ї через першу, тобто перевіряємо умову $в_i<в_1+д_{1i}$. У разі її невиконання замінюємо наявне значення $в_i$ на $в_1+д_{1i}$, а існуюче c_i - на 1.

Для елементів $в_2$ та $в_3$ збігаються значення, що порівнюються, тому елементам c_2 та c_3 припишемо два значення 0,1.

i	0	1	2	3	4	5	6	7
A_i	1	1	0	0	0	0	0	0
B_i	0	4	**5**	9	∞	6	∞	∞
C_i	0	0	0,1	0,1	0	1	0	0
2-й етап								
A_i	1	1	1	0	0	0	0	0
B_i	0	4	5	9	∞	**6**	16	∞
C_i	0	0	0,1	0,1	0	1	2	0
3-й етап								
A_i	1	1	1	0	0	1	0	0
B_i	0	4	5	**9**	14	6	16	21
C_i	0	0	0,1	0,1	5	1	2,5	5
4-й етап								
A_i	1	1	1	1	0	1	0	0
B_i	0	4	5	9	**14**	6	16	21
C_i	0	0	0,1	0,1	5	1	2,5	5
5-й етап								
A_i	1	1	1	1	1	1	0	0
B_i	0	4	5	9	14	6	**16**	21
C_i	0	0	0,1	0,1	5	1	2,5	5
6-й етап								
A_i	1	1	1	1	1	1	1	0
B_i	0	4	5	9	14	6	16	18
C_i	0	0	0,1	0,1	5	1	2,5	6

Серед елементів $в_i$ (для яких a_i=0). вибираємо найменший Це елемент $в_2$=5. Вершина 2 – поточна.

Поточна вершина 5.

Поточна вершина 5.

Поточна вершина 4.

Поточна вершина 6.

Для розгляду залишилась одна вершина x_7. Її вибір не змінить отриманих результатів.

Остання таблиця містить інформацію про довжину найкоротшого шляху від початкової вершини x_0 до будь-якої x_i. Сам шлях можна відтворити за елементами вектора C. Випишемо результат для вершини x_7.

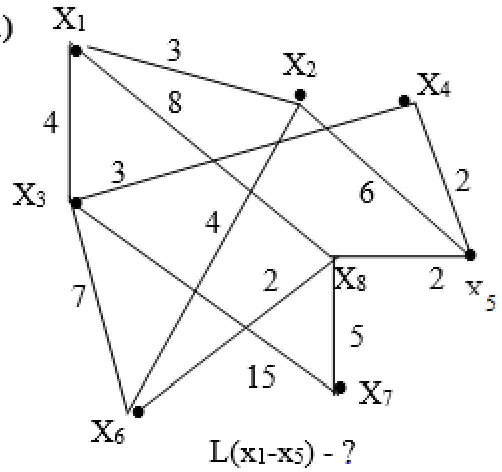
$L_{min}(x_0, x_7)=18$,

$$\mu_7^1 = \{x_0, x_2, x_6, x_7\},\ \mu_7^2 = \{x_0, x_1, x_2, x_6, x_7\},\ \mu_7^3 = \{x_0, x_1, x_5, x_6, x_7\}$$

Завдання № 3

Задано граф G. Зобразити граф у вигляді матриць суміжності й суміжності ваг. Знайти найкоротший шлях і його довжину між двома зазначеними вершинами графа

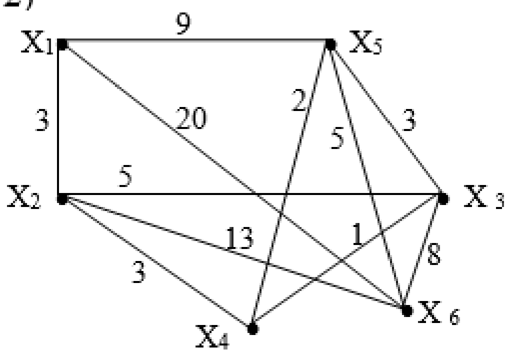
1)



L(x1-x5) - ?

M5min

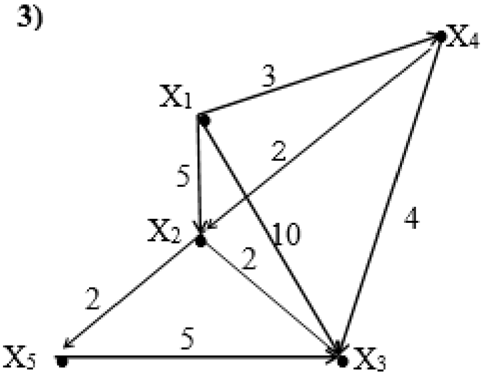
2)



μ^6_{min} - ?

L (x1-x6)- ?

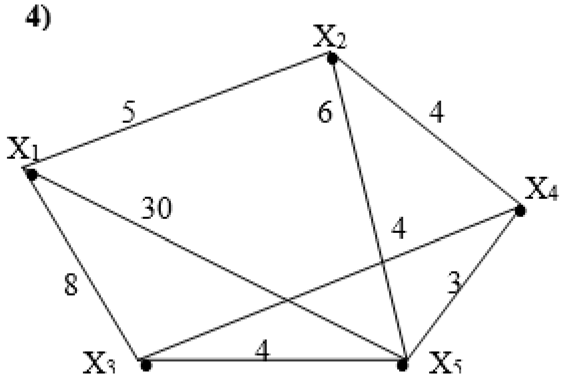
3)



L (x1-x3) - ?

μ^3 - ?

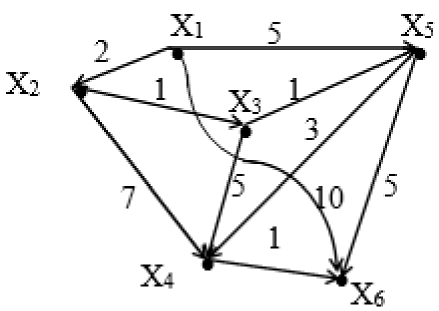
4)



L(x1-x5) - ?

μ^5 - ?

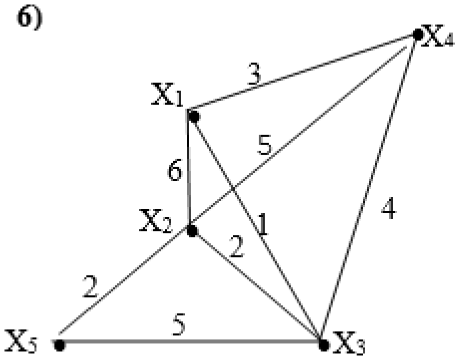
5)



L(x1-x6) - ?

μ^6_{min} - ?

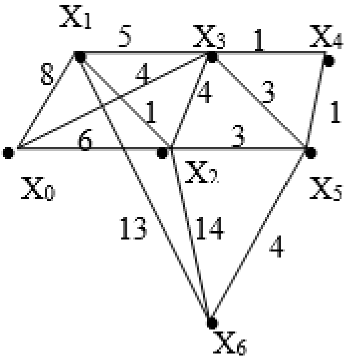
6)



L (x1-x5) - ?

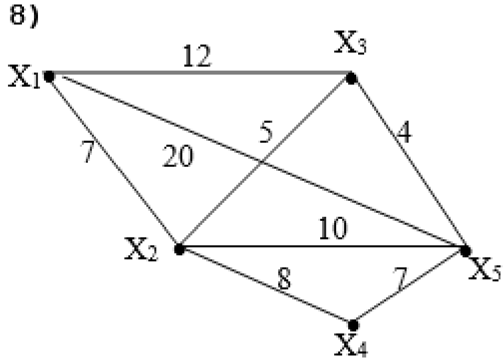
μ^5 - ?

7)



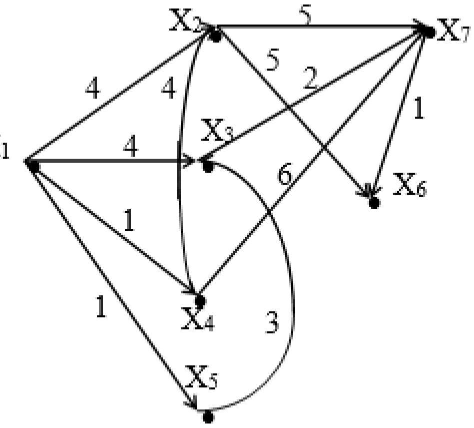
μ^6_{min} - ?
$L(x_0\text{-}x_6)$ - ?

8)



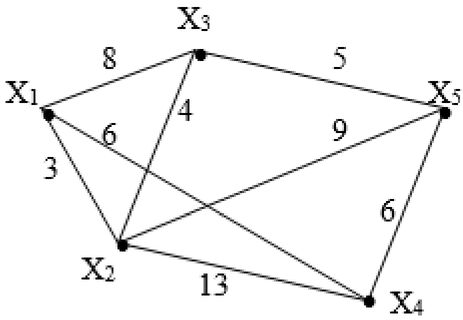
$L(x_1\text{-}x_5)$ -?
μ^5 - ?

9)



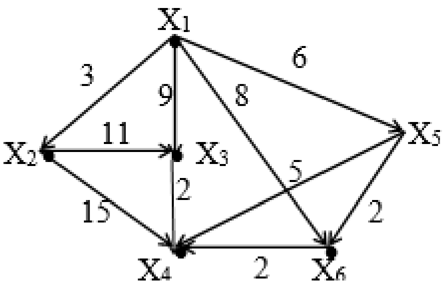
$L(x_1\text{-}x_7)$ - ?
μ^7_{min} - ?

10)



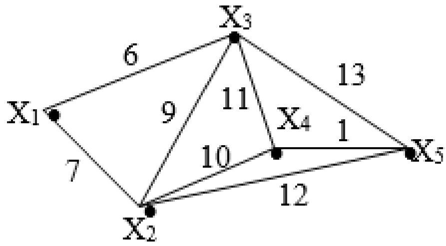
$L(x_1\text{-}x_5)$ - ?
μ^5_{min} -?

11)



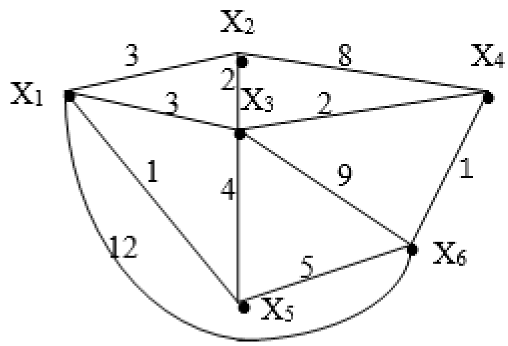
$L(x_1\text{-}x_4)$ - ?
μ^4_{min} - ?

12)



$L(x_1\text{-}x_5)$ - ?
μ^5_{min} - ?

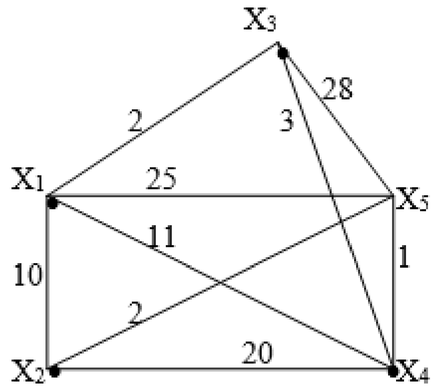
13)



L(x_1-x_6) - ?

μ^6_{min} - ?

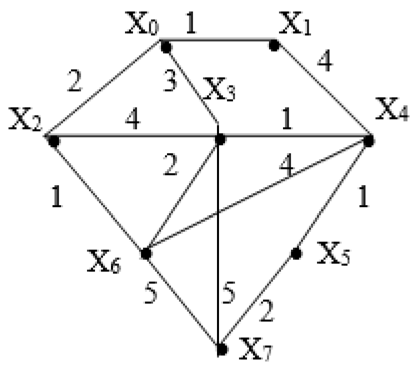
14)



L(x_1-x_5) - ?

μ^5 - ?

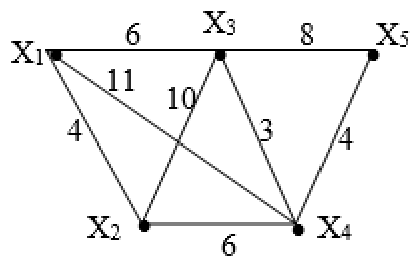
15)



μ^7_{min} - ?

L(x_0-x_7) - ?

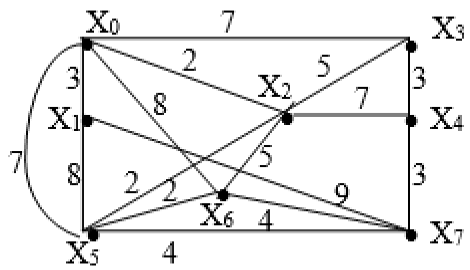
16)



L(x_1-x_5) - ?

μ^5 - ?

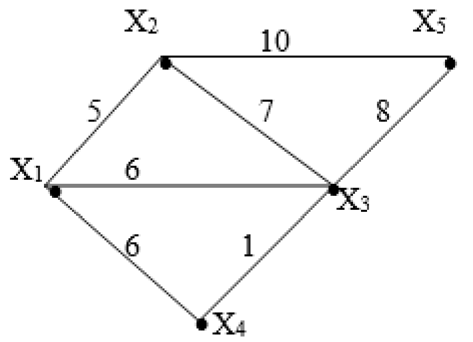
17)



μ^7_{min} - ?

L(x_0-x_7) - ?

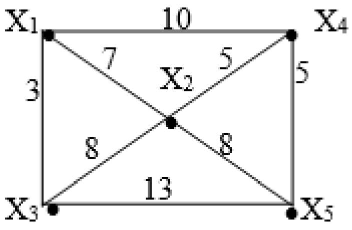
18)



L(x_1-x_5) - ?

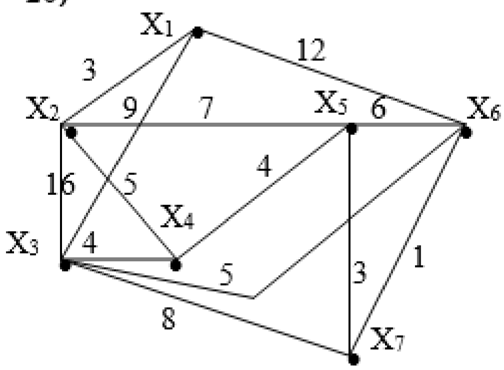
μ^5 - ?

19)



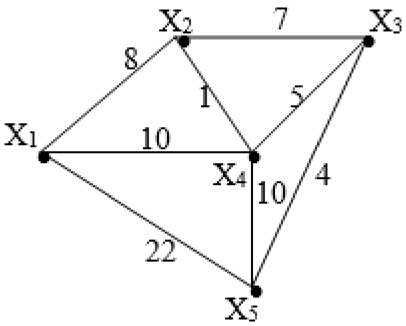
$L(x_1\text{-}x_5)$ - ?
μ^5_{min} - ?

20)



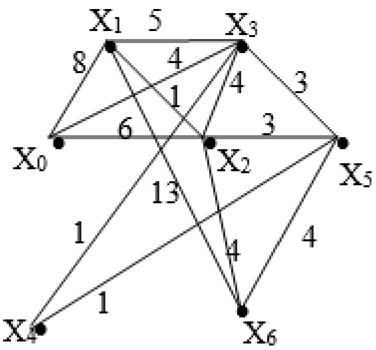
μ^6_{min} - ?
$L(x_2\text{-}x_6)$ - ?

21)



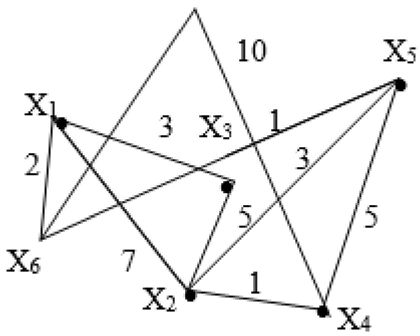
$L(x_1\text{-}x_5)$-?
μ^5_{min} -?

22)



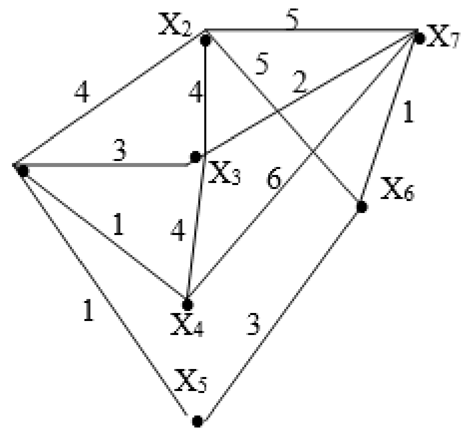
μ^6_{min} - ?
$L(x_0\text{-}x_6)$ - ?

23)



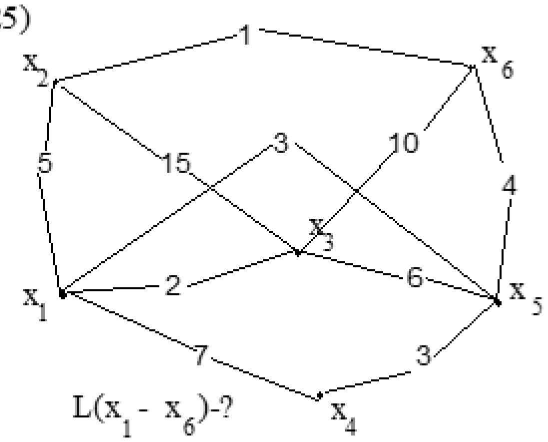
$L(x_1\text{-}x_4)$ - ?
μ^4_{min} - ?

24)



$L(x_1\text{-}x_7)$ - ?
μ^7_{min} - ?

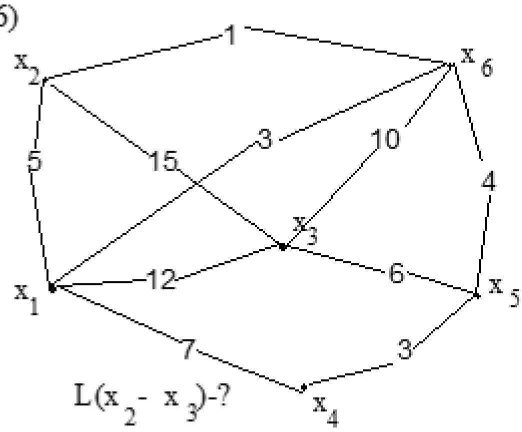
25)



$L(x_1 - x_6)$-?

μ_6-?

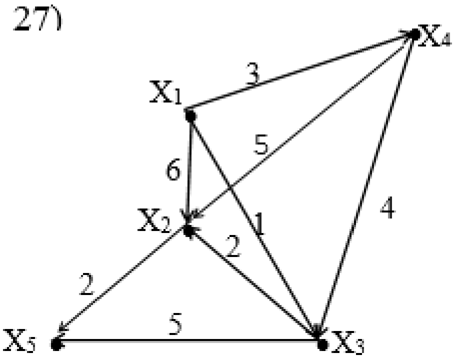
26)



$L(x_2 - x_3)$-?

μ_3-?

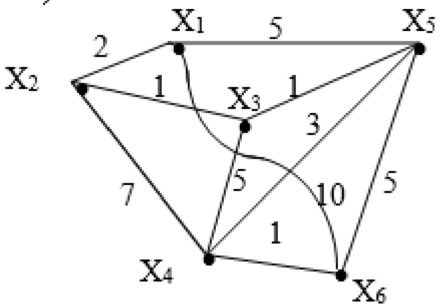
27)



L $(x_1$-$x_5)$ - ?

μ^5 - ?

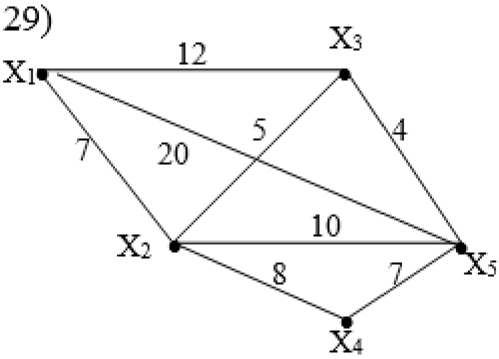
28)



$L(x_1$-$x_6)$ - ?

μ^6_{min} - ?

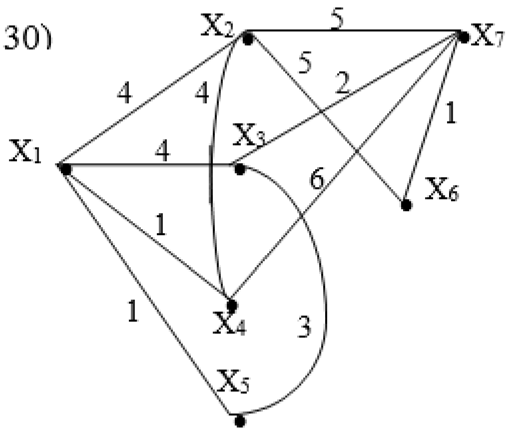
29)



$L(x_1$-$x_5)$ -?

μ^5 - ?

30)



$L(x_1$-$x_6)$ - ?

μ^6_{min} - ?

2.2 Пошук найкоротшого шляху між будь-якими двома вершинами графа матричним методом

Матричні методи дозволяють знаходити довжини найкоротших шляхів, але не можуть відтворити найкоротші шляхи, оскільки перелік вершин, які об'єднуються у шлях, губляться під час перерахунку матриці суміжності ваг. Серед матричних методів розглянемо метод Шимбела.

Шимбелом розроблений метод, що використовує спеціальні операції над елементами вихідної матриці суміжності ваг заданого графа, а саме виконується піднесення матриці до степеня з перетворенням основних операцій множення елементів та додавання отриманих добутків так:

– операція множення двох елементів a і b під час піднесення матриці до степеня, що відповідає їх алґебраїчній сумі, тобто a+b=b+a заміняємо на a·b=b·a;

– сума двох елементів у разі звичайного піднесення матриці до степеня замінюється вибором мінімального елемента, тобто a+b=min(a, b).

За допомогою зазначених операцій довжини найкоротших шляхів між вузлами мережі визначаються піднесенням матриці суміжності ваг до степеня.

Як основну матрицю для перетворень необхідно взяти зображення графа за допомогою матриці суміжності ваг, у якої по діагоналі занесено нулі. Піднесення матриці до степеня заключається у множенні матриць за правилами Шимбела. Нагадуємо, що для отримання елементу a_{ij} у новій матриці слід обрати для обчислень рядок першої матриці за номером i та стовпець другої матриці за номером j.

Тобто для отримання матриці А у другому степені за Шимбелом необхідно виконати такі дії:

$$A^2 = A \cdot A = \begin{pmatrix} 0 & a_{12} & a_{13} \\ a_{21} & 0 & a_{23} \\ a_{31} & a_{32} & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & a_{12} & a_{13} \\ a_{21} & 0 & a_{23} \\ a_{31} & a_{32} & 0 \end{pmatrix} = \begin{pmatrix} 0 & a'_{12} & a'_{13} \\ a'_{21} & 0 & a'_{23} \\ a'_{31} & a'_{32} & 0 \end{pmatrix},$$

де
$$\begin{aligned}
a'_{12} &= \min\{0 + a_{12};\, a_{12} + 0;\, a_{13} + a_{32}\};\\
a'_{13} &= \min\{0 + a_{13};\, a_{12} + a_{23};\, a_{13} + 0\};\\
a'_{21} &= \min\{a_{21} + 0;\, 0 + a_{21};\, a_{23} + a_{31}\};\\
a'_{23} &= \min\{a_{21} + a_{13};\, 0 + a_{23};\, a_{23} + 0\};\\
a'_{31} &= \min\{a_{31} + 0;\, a_{32} + a_{21};\, 0 + a_{31}\};\\
a'_{32} &= \min\{a_{31} + a_{12};\, a_{32} + 0;\, 0 + a_{32}\}.
\end{aligned}$$

Для отримання матриці A у третьому степені за Шимбелом необхідно виконати такі дії:

$$A^3 = A \cdot A^2 = \begin{pmatrix} 0 & a_{12} & a_{13} \\ a_{21} & 0 & a_{23} \\ a_{31} & a_{32} & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & a'_{12} & a'_{13} \\ a'_{21} & 0 & a'_{23} \\ a'_{31} & a'_{32} & 0 \end{pmatrix} = \begin{pmatrix} 0 & a''_{12} & a''_{13} \\ a''_{21} & 0 & a''_{23} \\ a''_{31} & a''_{32} & 0 \end{pmatrix},$$

де
$$\begin{aligned}
a''_{12} &= \min\{0 + a'_{12};\, a_{12} + 0;\, a_{13} + a'_{32}\};\\
a''_{13} &= \min\{0 + a'_{13};\, a_{12} + a'_{23};\, a_{13} + 0\};
\end{aligned}$$
і т.д.

Елементи матриці A^t визначають довжини найкоротших шляхів (ланцюгів), що складаються не більш ніж із t дуг (ребер). Якщо в процесі множення виявиться, що $A^{t-1}=A^t$, то це говорить про те, що довжини мінімальних шляхів (між будь-якими двома вершинами графа), що складаються не більш ніж із t-1 дуг (ребер), збігаються з довжинами шляхів (ланцюгів), що складаються не більш ніж із t дуг (ребер). Отже, подальше множення матриць немає сенсу, остільки не призводить до зміни елементів матриці A^{t-1}, і довжини найкоротших шляхів об’єднано у матрицю A^{t-1}.

Приклад 2.3 Нехай граф подано у вигляді матриці суміжності ваг $C = \begin{pmatrix} \infty & 3 & 4 & 5 \\ \infty & \infty & \infty & 1 \\ 4 & 2 & \infty & 1 \\ \infty & 8 & 2 & \infty \end{pmatrix}$. Необхідно визначити найкоротшу відстань між будь-якими двома вершинами графа.

Розв'язання. Будемо виконувати піднесення матриці суміжності ваг до степеня до повторення елементів двох матриць піднесення.

$$A = \begin{pmatrix} 0 & 3 & 4 & 5 \\ \infty & 0 & \infty & 1 \\ 4 & 2 & 0 & 1 \\ \infty & 8 & 2 & 0 \end{pmatrix},\ A^2 = A \cdot A = \begin{pmatrix} 0 & 3 & 4 & 4 \\ \infty & 0 & 3 & 1 \\ 4 & 2 & 0 & 1 \\ 6 & 4 & 2 & 0 \end{pmatrix},\ A^3 = A \cdot A^2 = \begin{pmatrix} 0 & 3 & 4 & 5 \\ 7 & 0 & 3 & 1 \\ 4 & 2 & 0 & 1 \\ 6 & 4 & 2 & 0 \end{pmatrix},$$

$$A^4 = A \cdot A^3 = A^2 \cdot A^2 = \begin{pmatrix} 0 & 3 & 4 & 4 \\ 7 & 0 & 3 & 1 \\ 4 & 2 & 0 & 1 \\ 6 & 4 & 2 & 0 \end{pmatrix}.$$

Як бачимо, матриці A^3 і A^4 збігаються. Отже, підсумковою матрицею найкоротших відстаней між двома будь-якими вершинами графа будемо вважати матрицю A^3, степінь якої говорить, що будь-який найкоротший шлях (ланцюг) утворюють не більш ніж 3 дуги (ребра).

Завдання № 4

Орієнтований граф G=(X,U) задано матрицею суміжності ваг C.

Необхідно :

– зобразити граф графічно й у вигляді матриці інциденцій;

– визначити хроматичне число графа;

– використовуючи метод Шимбела, визначити найкоротші відстані між будь-якими двома вершинами графа.

1. $C = \begin{pmatrix} \infty & 2 & 2 & \infty & 2 \\ 2 & \infty & \infty & \infty & 1 \\ \infty & 1 & \infty & 1 & 3 \\ 1 & \infty & 2 & \infty & 1 \\ \infty & 2 & 5 & 3 & \infty \end{pmatrix}$ 2. $C = \begin{pmatrix} \infty & 3 & \infty & 3 & 1 & 2 \\ 1 & \infty & 3 & 1 & 5 & \infty \\ 4 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 4 & \infty & 4 & 1 \\ \infty & 3 & 3 & 2 & \infty & 3 \\ \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$ 3. $C = \begin{pmatrix} \infty & 5 & 2 & 1 & 2 \\ 2 & \infty & 5 & 8 & 1 \\ 5 & 1 & \infty & 1 & 3 \\ 1 & 5 & 2 & \infty & 1 \\ 5 & 2 & 5 & 5 & \infty \end{pmatrix}$

4. $C=\begin{pmatrix} \infty & 3 & \infty & \infty & 6 & 4 \\ \infty & 5 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 2 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & 4 & 10 & 2 & \infty & \infty \\ \infty & 1 & \infty & \infty & 4 & \infty \end{pmatrix}$ 5. $C=\begin{pmatrix} \infty & \infty & 2 & 1 & 2 \\ 2 & \infty & \infty & \infty & 1 \\ \infty & 1 & \infty & 1 & 3 \\ 1 & \infty & 2 & \infty & \infty \\ \infty & 2 & 4 & \infty & \infty \end{pmatrix}$ 6. $C=\begin{pmatrix} \infty & 3 & 4 & \infty & \infty \\ \infty & \infty & 2 & \infty & \infty \\ \infty & \infty & \infty & 5 & 3 \\ 1 & \infty & \infty & \infty & \infty \\ \infty & 6 & 2 & \infty & \infty \end{pmatrix}$

7. $C=\begin{pmatrix} \infty & 5 & 2 & \infty & \infty \\ 2 & \infty & 4 & 10 & 8 \\ 1 & 4 & \infty & 3 & 3 \\ \infty & \infty & 2 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{pmatrix}$ 8. $C=\begin{pmatrix} \infty & 3 & \infty & \infty & \infty & \infty \\ \infty & \infty & 3 & \infty & \infty & \infty \\ 1 & \infty & \infty & \infty & \infty & \infty \\ 5 & \infty & \infty & \infty & 3 & \infty \\ \infty & \infty & 2 & \infty & \infty & 1 \\ \infty & 4 & 6 & 4 & \infty & \infty \end{pmatrix}$ 9. $C=\begin{pmatrix} \infty & 2 & \infty & 1 & \infty \\ 1 & \infty & 2 & 5 & 1 \\ \infty & \infty & \infty & \infty & 2 \\ \infty & 4 & \infty & \infty & 1 \\ \infty & \infty & 2 & 1 & \infty \end{pmatrix}$

10. $C=\begin{pmatrix} \infty & 1 & \infty & 2 & 1 \\ \infty & \infty & \infty & 1 & 1 \\ 3 & \infty & \infty & \infty & 2 \\ \infty & 6 & 2 & \infty & 8 \\ \infty & \infty & 3 & \infty & \infty \end{pmatrix}$ 11. $C=\begin{pmatrix} \infty & 1 & 2 & \infty & 3 \\ \infty & \infty & 3 & 4 & \infty \\ 2 & \infty & \infty & \infty & 1 \\ \infty & 2 & \infty & \infty & 2 \\ 2 & \infty & 1 & 2 & \infty \end{pmatrix}$ 12. $C=\begin{pmatrix} \infty & 8 & 2 & 3 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & 5 & \infty & 2 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 4 & 9 & \infty & 3 \\ \infty & 6 & 2 & \infty & \infty & \infty \end{pmatrix}$

13. $C=\begin{pmatrix} \infty & \infty & 2 & 1 & 2 \\ \infty & \infty & 2 & 5 & 2 \\ \infty & 2 & \infty & 1 & \infty \\ 1 & 3 & \infty & \infty & \infty \\ 5 & 2 & \infty & \infty & \infty \end{pmatrix}$ 14. $C=\begin{pmatrix} \infty & 2 & 2 & 1 & 2 \\ \infty & \infty & \infty & \infty & 1 \\ \infty & 4 & \infty & 1 & \infty \\ 1 & \infty & 2 & \infty & 1 \\ 1 & 2 & 4 & \infty & \infty \end{pmatrix}$ 15. $C=\begin{pmatrix} \infty & 4 & \infty & 2 & \infty & \infty \\ \infty & \infty & 3 & \infty & \infty & \infty \\ \infty & \infty & \infty & 4 & 3 & 8 \\ \infty & \infty & \infty & \infty & 5 & 2 \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$

16. $C=\begin{pmatrix} \infty & \infty & 4 & \infty & 1 & 2 \\ 1 & \infty & 3 & 1 & \infty & \infty \\ 4 & 5 & \infty & 3 & \infty & \infty \\ \infty & \infty & 2 & \infty & 4 & 1 \\ \infty & 3 & 3 & 2 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$ 17. $C=\begin{pmatrix} \infty & \infty & 2 & \infty & 2 \\ \infty & \infty & 2 & 18 & 1 \\ 5 & \infty & \infty & 6 & 3 \\ 1 & 5 & 2 & \infty & \infty \\ \infty & 2 & \infty & 5 & \infty \end{pmatrix}$ 18. $C=\begin{pmatrix} \infty & 2 & \infty & 5 & \infty & \infty \\ \infty & \infty & 3 & 2 & \infty & \infty \\ 4 & \infty & \infty & 5 & \infty & \infty \\ \infty & 3 & 6 & \infty & \infty & \infty \\ 4 & \infty & 2 & \infty & \infty & \infty \\ \infty & 8 & \infty & 8 & 3 & \infty \end{pmatrix}$

19. $C=\begin{pmatrix} \infty & \infty & 2 & 1 & 2 \\ 2 & \infty & \infty & 8 & 1 \\ \infty & 1 & \infty & 1 & 3 \\ 1 & \infty & 2 & \infty & 1 \\ \infty & 2 & 5 & 5 & \infty \end{pmatrix}$ 20. $C=\begin{pmatrix} \infty & \infty & 4 & \infty & 3 & 2 \\ 1 & \infty & 3 & 1 & \infty & \infty \\ 4 & 5 & \infty & 3 & \infty & \infty \\ \infty & \infty & \infty & \infty & 3 & 2 \\ \infty & 3 & 2 & 3 & \infty & \infty \\ \infty & 2 & \infty & 1 & \infty & \infty \end{pmatrix}$ 21. $C=\begin{pmatrix} \infty & 1 & 4 & \infty & 1 \\ \infty & \infty & \infty & 1 & 1 \\ 3 & \infty & \infty & 3 & 2 \\ \infty & 6 & 2 & \infty & \infty \\ \infty & \infty & 3 & \infty & \infty \end{pmatrix}$

22. $C = \begin{pmatrix} \infty & 2 & 4 & \infty & \infty & 2 \\ 1 & \infty & \infty & 1 & \infty & \infty \\ 3 & \infty & \infty & 3 & 4 & \infty \\ \infty & 1 & 2 & \infty & 3 & 1 \\ \infty & 3 & 3 & \infty & \infty & 5 \\ 1 & \infty & \infty & 1 & \infty & \infty \end{pmatrix}$ 23. $C = \begin{pmatrix} \infty & 4 & \infty & 4 & 2 & 3 \\ 2 & \infty & 4 & 2 & 6 & \infty \\ 5 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 5 & \infty & 5 & 1 \\ \infty & 1 & 2 & 3 & \infty & 1 \\ \infty & 1 & \infty & 4 & \infty & \infty \end{pmatrix}$ 24. $C = \begin{pmatrix} \infty & 5 & 2 & 1 & \infty \\ 2 & \infty & 6 & 9 & 1 \\ 6 & 1 & \infty & 1 & 3 \\ 1 & 5 & 2 & \infty & 2 \\ 5 & 2 & \infty & 6 & \infty \end{pmatrix}$

25. $C = \begin{pmatrix} \infty & 5 & 2 & \infty & \infty \\ 2 & \infty & 4 & 10 & 8 \\ 1 & 4 & \infty & 3 & 3 \\ \infty & \infty & 2 & \infty & 1 \\ 1 & \infty & 2 & \infty & \infty \end{pmatrix}$ 26. $C = \begin{pmatrix} \infty & 3 & \infty & \infty & \infty & 1 \\ \infty & \infty & 3 & \infty & 1 & \infty \\ 1 & \infty & \infty & \infty & \infty & 1 \\ 5 & \infty & 1 & \infty & 3 & \infty \\ \infty & \infty & 2 & \infty & \infty & 1 \\ \infty & 4 & 6 & 4 & \infty & \infty \end{pmatrix}$ 27. $C = \begin{pmatrix} \infty & 2 & 1 & 1 & \infty \\ 1 & \infty & 2 & 7 & 1 \\ \infty & 1 & \infty & \infty & 2 \\ \infty & 8 & \infty & \infty & 6 \\ 1 & \infty & 2 & 1 & \infty \end{pmatrix}$

28. $C = \begin{pmatrix} \infty & 1 & \infty & 2 & 1 \\ \infty & 1 & \infty & 1 & 1 \\ 3 & \infty & \infty & \infty & 2 \\ \infty & 1 & 2 & \infty & 8 \\ 1 & \infty & 3 & \infty & \infty \end{pmatrix}$ 29. $C = \begin{pmatrix} \infty & 1 & 2 & \infty & 3 \\ 1 & \infty & 3 & 8 & \infty \\ 2 & \infty & \infty & 1 & 1 \\ \infty & 2 & \infty & \infty & 2 \\ 2 & \infty & 1 & 12 & \infty \end{pmatrix}$ 30. $C = \begin{pmatrix} \infty & 8 & 2 & 3 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & 5 & \infty & 2 & \infty & 1 \\ \infty & \infty & \infty & \infty & \infty & \infty \\ 28 & \infty & 4 & 10 & \infty & 3 \\ \infty & 1 & 2 & \infty & 1 & \infty \end{pmatrix}$

2.3 Пошук найкоротшого гамільтонова контуру методом гілок та меж

Загальна задача комівояжера полягає у такому: використовуючи систему транспортних сполучень (доріг і т. д.) між пунктами (містами, фірмами і т. д.) у конкретній зоні обслуговування, відвідати всі пункти у такій послідовності, щоб пройдений маршрут був найкоротшим із усіх можливих.

На мові теорії графів загальна задача комівояжера має таке формулювання: у зваженому зв’язному графі G знайти найкоротший маршрут, що проходить через усі вершини графа. Ураховуючи необхідність повернення комівояжера у пункт вихідного перебування, у постановці задачі можлива додаткова вимога замкненості маршруту комівояжера. Для довільного графа загальна задача комівояжера має розв’язання.

Але існує ще одна постановка задачі, у якій додатково до попередньої задачі треба, щоб кожний пункт обслуговування комівояжер відвідав тільки один раз. У такій постановці задача не завжди має розв’язання, а якщо і має, то маршрут комівояжера є найкоротшим гамільтоновим ланцюгом або циклом. Гамільтоновим ланцюгом у графі G=(X,U) називається простий шлях, що містить усі вершини графа. Якщо верщина відправлення гамільтонового ланцюгу збігається з вершиною прибуття, то такий шлях слід називати гамільтоновим контуром.

Поширеною є така типова ситуація. Є технічний пристрій (верстат, комп'ютер і т. д.) і n операцій (задач), кожну з яких пристрій здатний здійснити (розв'язати після виконання відповідної перенастройки (перепрограмування). Точніше під час переходу від i-ої задачі до j-ої задачі перепрограмування пристрою потребує витратити t_{ij} одиниць часу або іншого ресурсу. Необхідно знайти послідовність виконання завдань, за якої час кожного перепрограмування не перевищує заданого значення τ. У цьому випадку будується зв'язний граф, у якому дуга (i, j) наявна тоді й тільки тоді, коли $t_{ij} \le \tau$. Будь-який гамільтонів шлях задає шукану послідовність виконання завдань.

Побудуємо математичну модель «гамільтонової» задачі комівояжера (торгівельного аґента).

Нехай $x_{ij} = \begin{cases} 1, \text{якщо комівояжер з } i-\text{го міста їде до } j-\text{го} \\ 0, \quad \text{в іншому випадку} \end{cases}$ де i,j=1,2,…,n; i≠j.

Необхідно мінімізувати функцію витрат $F(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$.

Якщо будуть виконані такі умови: $\sum_{i=1}^{n} x_{ij} = 1, j = 1,2,..,n \quad \sum_{j=1}^{n} x_{ij} = 1, i = 1,2,..,n$

$$u_i - u_j + nx_{ij} \le n-1, i, j = 1,2,\dots,n \quad i \ne j$$

де u_i та u_j – довільні значення.

Історично першим розглядав задачу обходу простим циклом усіх вершин графа у 1859 року відомий ірландський математик У. Гамільтон. Він побудував низку таких циклів на ребрах додекаедра – правильного опуклого багатогранника з 20 вершинами і 12 п'ятикутними гранями. Гамільтон трактував свою задачу у вигляді гри «навколосвітня подорож», у якій пропонується вибрати маршрут відвідування двадцяти міст на земній кулі, рухаючись по дорогах – ребрах додекаедра. Він навіть продав свою гру торговцеві іграшками за 25 гіней.

На сьогодні існує безліч прямих та наближених методів розв'язання задачі побудови гамільтонова контуру. Розгланемо методику розв'язання задачі методом гілок та меж, який досить ефективний для розв'язання описаної задачі на графі, що містить не більш ніж 40 вершин.

Основна ідея методу досить проста. Спочатку будується деяка оцінка знизу (НГ – нижня оцінка) довжин усіх гамільтонових контурів. Після цього множина всіх гамільтонових контурів розбивається на дві підмножини. Перша підмножина складається з гамільтонових контурів, що містять деяку дугу (i,j) , а інша підмножина складається з контурів, що не містять цієї дуги $(\overline{i,j})$. Для кожної з підмножини за тим самим правилом, що і для початкової множини гамільтонових маршрутів, визначається нижня межа. Кожна нова нижня межа виявляється не менша за нижню межу, визначену для всієї множини. Порівнюючи нижні межі, можна визначити підмножину гамільтонових контурів, у середині якої з більшою ймовірністю міститься оптимальний маршрут. Ця підмножина за анологічним правилом розбивається ще на два, і знову знаходяться нижні межі, і так доти, поки не залишається єдиний цикл. Процес розбивки підмножин супроводжується побудовою деякого бінарного дерева .

Отримавши деякий гамільтонів контур, переглядають обірвані гілки дерева, і якщо нижні межі підмножин, що віднесені обірваним гілкам, виявляються меншими, ніж довжина знайденого контуру, то ці гілки розвивають за тим самим правилом, поки не отримаємо маршруту меншої довжини або не переконаємося, що серед цих підмножин не може міститися подібного маршруту. Ті ж гілки, для яких нижня оцінка виявиться більшою, ніж отриманий маршрут, виключають із розгляду.

Ідея одержання нижньої оцінки пов'язана з тим, що якщо розв'язати задачу про комівояжера з деякою матрицею відстаней, а потім із якого-небудь рядка або стовпця цієї матриці відняти довільне позитивне число, то розв'язок задачі про комівояжера з цією зміненою матрицею відстаней збігається з колишнім розв'язком, а довжина маршруту змінюється на це ж саме число. Якщо цю операцію проробити і для інших рядків і стовпців, то довжина маршруту буде відрізнятися на суму всіх чисел, що віднімаються з рядків і стовпців.

Розглянемо розв'язання цієї задачі на *прикладі 2.4.* Нехай необхідно розв'язати задачу пошуку гамільтонова контура для випадку 4-х міст. Як відомо, кількість будь-яких гамільтонових контурів дорівнює (n-1)!, тобто (3)!=6. Матриця витрат має вигляд $C = \begin{pmatrix} \infty & 15 & 5 & 2 \\ 10 & \infty & 3 & 20 \\ 16 & 8 & \infty & 5 \\ 2 & 3 & 2 & \infty \end{pmatrix}$.

Розв`язання. Виконаємо крок приведення матриці до вигляду: у кожному рядку та стовпці необхідно отримати нуль, що дозволить виявити ребра графа з найменшою вагою.

	1	2	3	4	h_i
1	∞	15	5	2	2
2	10	∞	3	20	3
3	16	8	∞	5	5
4	2	3	2	∞	2
					12

	1	2	3	4	
1	∞	13	3	0	
2	7	∞	0	17	
3	11	3	∞	0	
4	0	1	0	∞	
h_j	0	1	0	0	1

	1	2	3	4
1	∞	12	3	0^3
2	7	∞	0^7	17
3	11	2	∞	0^2
4	0^7	0^2	0^0	∞

Сума всіх констант приведення матриці дорівнює НМ=12+1=13. Отримали початкове значення нижньої межі. До кожного нуля матриці припишемо вагу (штраф за невикористання проїзду), яка складається із суми найменших елементів із стовпця та рядка, на перетині яких знаходиться нуль (нуль, що оцінюється, у виборі мінімального не фігурує). Наприклад, для нуля (1,4) min(рядка)=3 min(стовпця)=0, тоді вага нуля дорівнює 3+0=3. Для подальшого розгляду вибираємо нуль з найбільшою вагою, щоб уникнути можливого штрафу за невикористання проїзду. Якщо таких нулів декілька, вибираємо будь-який. Наприклад, нуль (2,3). Отже, будемо аналізувати можливе включення в підсумковий Гамільтонів контур проїзду з 2-го міста до 3-го за прямим шляхом.

У разі виключення такого проїзду необхідно заборонити елемент матриці (2,3) для вибору, тобто замінити його на нескінченність, що призведе до повторного приведення рядка матриці на 7 одиниць (вага нуля). Тоді нижня межа такого варіанту дорівнює НМ=НМ+3=13+7=20. Для аналізу такого варіанту немає потреби виписувати матрицю витрат, достатньо врахувати під час підрахунку нижньої межі вагу вибраного нуля матриці.

Розглянемо можливе включення проїзду з 2-го до 3-го міста. З матриці необхідно викреслити 2-й рядок (щоб виключити можливий переїзд із 2-го міста в будь-яке інше, окрім 3-го) та 3-й стовпець (щоб виключити можливий переїзд до 3-го міста з будь-якого іншого, окрім 2-го), а також заборонити до вибору елемент матриці (3,2) (щоб виключити повернення агента в місто відправлення, оскільки він ще не потрапив до всіх запланованих міст). Над перетвореною матрицею виконаємо крок приведення матриці (якщо це необхідно). Тоді нижня межа цього вибору буде дорівнювати сумі констант приведення перетвореної матриці та попереднього значення нижньої межі.

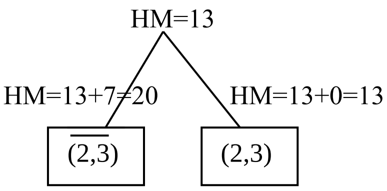
	1	2	3	4
1	∞	12	3	0^{3}
2	7	∞	0^{7}	17
3	11	2	∞	0^{2}
4	0^{7}	0^{2}	0^{0}	∞

	1	2	4	h_i
1	∞	12	0	0
3	11	∞	0	0
4	0	0	∞	0
	0	0	0	0

(2,3)	1	2	4
1	∞	12	0^{12}
3	11	∞	0^{11}
4	0^{11}	0^{12}	∞

НМ=13+0=13.

На цьому етапі маємо дерево пошуку такого вигляду:

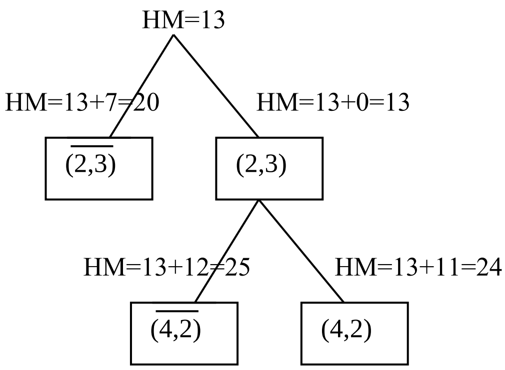


Бачимо, що найменшого значення набуває НМ для варіанту проїзду з 2-го міста до 3-го. Залишимося на цьому варіанті. Продовжимо аналіз у матриці (2,3). Знову до кожного нуля припишемо його вагу і виберемо для розгляду нуль з найбільшою вагою. Наприклад, нуль (4,2). Варіант заперечення включення подібного переїзду «коштує» НМ=13+12=25. Варіант включення дуги (4,2) виключить з матриці 4-й рядок і 2-й стовпець. Дуга (4,2) з раніше проаналізованою дугою (2,3) створюють зв’язний шлях {4,2,3}. Тому потрібно заборонити можливе повернення з 3-го міста до 4-го, а саме – замінити елемент матриці (3,4) на нескінченність.

	1	2	4
1	∞	12	0^{12}
3	11	∞	0^{11}
4	0^{11}	0^{12}	∞

(4,2)	1	4	h_i
1	∞	0	0
3	11	∞	11
			11

(4,2)	1	4
1	∞	^{0}
3	0	∞



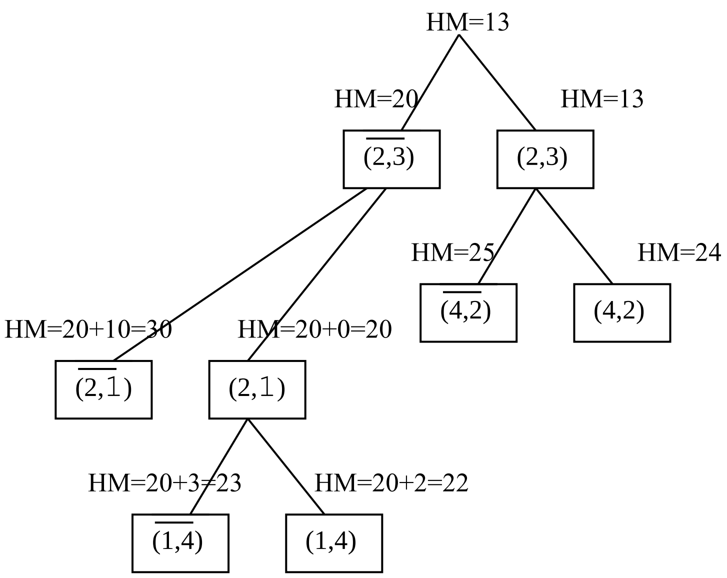
Після приведення матриці отримуємо НМ=НМ+11=13+11=24. Дерево пошуку має вигляд. Аналізуючи дерево пошуку, дійдемо висновку, що варіант виключення (2,3) найкращий. Необхідно повернутись до вихідної матриці витрат з оцінкою всіх нулів, замінити нуль (2,3) на нескінченність та привести матрицю. У результаті отримаємо матрицю $\overline{(2,3)}$. До кожного нуля приписуємо вагу та вибираємо найважчий, тобто нуль (2,1).

	1	2	3	4
1	∞	12	3	0^3
2	0^{10}	∞	∞	10
3	11	2	∞	0^2
4	0^0	0^2	0^3	∞

(2,1)	2	3	4	h_i
1	∞	3	0	0
3	2	∞	0	0
4	0	0	∞	0
	0	0	0	0

(2,1)	2	3	4
1	∞	3	0^3
3	2	∞	0^2
4	0^2	0^3	∞

Варіант заперечення включення подібного переїзду «коштує» НМ=20+10=30. Варіант включення дуги (2,1) виключить з матриці 2-й рядок і 1-й стовпець, а також призведе до заміни елемента (1,2) на нескінченність. Нижня межа включення НМ=НМ+0=20. Виконуючи дії, аналогічні описаним, отримуємо результуюче дерево пошуку.



Серед обірваних гілок вибираємо найлегшу з НМ=22. Їй відповідає матриця (1,4).

(1,4)	2	3
3	0	∞
4	∞	0

Матриця (1,4) потребує автоматичного включення до результуючого гамільтонового контуру дуг (3,2) та (4,3). У результаті маємо найкоротший Гамільтонів контур 1→4→3→2→1 завдовжки (ціною) НМ=22 умовні одиниці.

Завдання № 5

Комівояжерові (торгівельному агентові) необхідно проїхати через n міст, відвідавши кожне з них тільки раз, і повернутися у вихідне місто. Знайти маршрут комівояжера, що мінімізує сумарні витрати на час проїзду. Час, необхідний комівояжерові для проїзду з кожного i-ого міста в кожне j-те дорівнює c_{ij} і зазначено в матриці витрат С. Матриці витрат відповідно до номера варіанту наведено нижче. Розв’язати задачу комівояжера методом гілок і меж.

1. $C = \begin{Vmatrix} \infty & 3 & 12 & 14 & 15 \\ 3 & \infty & 1 & 3 & 1 \\ 13 & 2 & \infty & 9 & 6 \\ 12 & 3 & 9 & \infty & 4 \\ 8 & 2 & 11 & 6 & \infty \end{Vmatrix}$ 2. $C = \begin{Vmatrix} \infty & 15 & 26 & 31 & 18 \\ 35 & \infty & 20 & 27 & 19 \\ 51 & 24 & \infty & 35 & 18 \\ 24 & 18 & 40 & \infty & 17 \\ 17 & 17 & 25 & 19 & \infty \end{Vmatrix}$ 3. $C = \begin{Vmatrix} \infty & 61 & 12 & 62 & 43 \\ 64 & \infty & 10 & 55 & 38 \\ 87 & 78 & \infty & 11 & 47 \\ 70 & 42 & 11 & \infty & 53 \\ 54 & 25 & 76 & 27 & \infty \end{Vmatrix}$

4. $C = \begin{Vmatrix} \infty & 40 & 56 & 29 & 12 \\ 20 & \infty & 29 & 13 & 15 \\ 30 & 25 & \infty & 45 & 11 \\ 36 & 32 & 40 & \infty & 14 \\ 13 & 24 & 13 & 12 & \infty \end{Vmatrix}$ 5. $C = \begin{Vmatrix} \infty & 3 & 2 & 4 & 0 & 0 \\ 0 & \infty & 0 & 6 & 3 & 2 \\ 6 & 0 & \infty & 8 & 10 & 12 \\ 10 & 0 & 4 & \infty & 6 & 11 \\ 12 & 0 & 8 & 9 & \infty & 10 \\ 1 & 4 & 0 & 0 & 3 & \infty \end{Vmatrix}$ 6. $C = \begin{Vmatrix} \infty & 7 & 27 & 12 & 10 \\ 6 & \infty & 8 & 5 & 7 \\ 3 & 11 & \infty & 12 & 3 \\ 10 & 12 & 9 & \infty & 0 \\ 20 & 20 & 19 & 14 & \infty \end{Vmatrix}$

7. $C = \begin{Vmatrix} \infty & 6 & 17 & 18 & 11 \\ 0 & \infty & 10 & 0 & 5 \\ 10 & 0 & \infty & 6 & 10 \\ 15 & 8 & 3 & \infty & 0 \\ 14 & 0 & 11 & 7 & \infty \end{Vmatrix}$ 8. $C = \begin{Vmatrix} \infty & 30 & 40 & 30 & 10 \\ 30 & \infty & 13 & 6 & 21 \\ 28 & 44 & \infty & 34 & 8 \\ 7 & 24 & 7 & \infty & 7 \\ 51 & 30 & 22 & 47 & \infty \end{Vmatrix}$ 9. $C = \begin{Vmatrix} \infty & 7 & 27 & 37 & 27 \\ 27 & \infty & 3 & 10 & 18 \\ 25 & 5 & \infty & 31 & 41 \\ 4 & 4 & 4 & \infty & 21 \\ 48 & 27 & 44 & 19 & \infty \end{Vmatrix}$

10. $C = \begin{Vmatrix} \infty & 6 & 0 & 10 & 21 \\ 5 & \infty & 8 & 12 & 21 \\ 0 & 8 & \infty & 9 & 20 \\ 10 & 5 & 9 & \infty & 15 \\ 8 & 7 & 0 & 0 & \infty \end{Vmatrix}$ 11. $C = \begin{Vmatrix} \infty & 30 & 46 & 19 & 10 \\ 2 & \infty & 19 & 3 & 21 \\ 2 & 15 & \infty & 35 & 26 \\ 10 & 22 & 30 & \infty & 3 \\ 4 & 14 & 13 & 12 & \infty \end{Vmatrix}$ 12. $C = \begin{Vmatrix} \infty & 4 & 12 & 14 & 4 \\ 3 & \infty & 12 & 10 & 10 \\ 12 & 4 & \infty & 9 & 5 \\ 3 & 11 & 11 & \infty & 9 \\ 10 & 3 & 1 & 3 & \infty \end{Vmatrix}$

13. $C = \begin{Vmatrix} \infty & 10 & 21 & 26 & 13 \\ 30 & \infty & 15 & 22 & 14 \\ 46 & 19 & \infty & 30 & 13 \\ 19 & 13 & 35 & \infty & 12 \\ 12 & 12 & 20 & 14 & \infty \end{Vmatrix}$ 14. $C = \begin{Vmatrix} \infty & 3 & 3 & 12 & 3 \\ 5 & \infty & 10 & 8 & 8 \\ 5 & 4 & \infty & 5 & 8 \\ 5 & 12 & 15 & \infty & 9 \\ 14 & 5 & 17 & 2 & \infty \end{Vmatrix}$ 15. $C = \begin{Vmatrix} \infty & 10 & 30 & 40 & 30 \\ 30 & \infty & 6 & 13 & 21 \\ 28 & 8 & \infty & 34 & 44 \\ 7 & 7 & 7 & \infty & 24 \\ 51 & 30 & 47 & 22 & \infty \end{Vmatrix}$

16. $C = \begin{Vmatrix} \infty & 3 & 11 & 3 & 9 \\ 2 & \infty & 3 & 11 & 2 \\ 10 & 12 & \infty & 11 & 0 \\ 12 & 10 & 8 & \infty & 2 \\ 12 & 10 & 4 & 5 & \infty \end{Vmatrix}$ 17. $C = \begin{Vmatrix} \infty & 2 & 6 & 5 & 3 & 5 \\ 0 & \infty & 5 & 7 & 10 & 8 \\ 1 & 4 & \infty & 4 & 3 & 6 \\ 1 & 7 & 3 & \infty & 3 & 8 \\ 8 & 9 & 0 & 0 & \infty & 0 \\ 13 & 10 & 7 & 3 & 3 & \infty \end{Vmatrix}$ 18. $C = \begin{Vmatrix} \infty & 27 & 37 & 27 & 7 \\ 27 & \infty & 10 & 3 & 18 \\ 25 & 41 & \infty & 31 & 5 \\ 4 & 21 & 4 & \infty & 4 \\ 48 & 27 & 19 & 44 & \infty \end{Vmatrix}$

19. $C=\begin{Vmatrix} \infty & 3 & 23 & 18 & 11 \\ 4 & \infty & 14 & 16 & 17 \\ 4 & 3 & \infty & 8 & 9 \\ 4 & 9 & 4 & \infty & 0 \\ 12 & 12 & 2 & 2 & \infty \end{Vmatrix}$ 20. $C=\begin{Vmatrix} \infty & 7 & 1 & 11 & 12 \\ 6 & \infty & 3 & 7 & 6 \\ 6 & 8 & \infty & 9 & 10 \\ 11 & 0 & 4 & \infty & 0 \\ 18 & 11 & 4 & 4 & \infty \end{Vmatrix}$ 21. $C=\begin{Vmatrix} \infty & 10 & 19 & 46 & 30 \\ 2 & \infty & 3 & 19 & 15 \\ 2 & 21 & \infty & 30 & 21 \\ 10 & 26 & 35 & \infty & 14 \\ 4 & 3 & 12 & 13 & \infty \end{Vmatrix}$

22. $C=\begin{Vmatrix} \infty & 1 & 10 & 35 & 21 \\ 2 & \infty & 3 & 20 & 18 \\ 2 & 12 & \infty & 25 & 30 \\ 6 & 26 & 35 & \infty & 14 \\ 4 & 3 & 12 & 15 & \infty \end{Vmatrix}$ 23. $C=\begin{Vmatrix} \infty & 5 & 25 & 35 & 25 \\ 25 & \infty & 1 & 8 & 16 \\ 23 & 3 & \infty & 29 & 39 \\ 2 & 2 & 2 & \infty & 19 \\ 46 & 25 & 42 & 17 & \infty \end{Vmatrix}$ 24. $C=\begin{Vmatrix} \infty & 10 & 21 & 26 & 13 \\ 30 & \infty & 15 & 22 & 14 \\ 46 & 19 & \infty & 30 & 13 \\ 19 & 13 & 35 & \infty & 12 \\ 12 & 12 & 20 & 14 & \infty \end{Vmatrix}$

25. $C=\begin{Vmatrix} \infty & 11 & 20 & 47 & 31 \\ 3 & \infty & 4 & 20 & 16 \\ 3 & 22 & \infty & 31 & 22 \\ 11 & 27 & 36 & \infty & 15 \\ 5 & 4 & 13 & 14 & \infty \end{Vmatrix}$ 26. $C=\begin{Vmatrix} \infty & 51 & 2 & 40 & 43 \\ 44 & \infty & 1 & 30 & 25 \\ 45 & 56 & \infty & 1 & 17 \\ 25 & 30 & 2 & \infty & 25 \\ 4 & 3 & 4 & 3 & \infty \end{Vmatrix}$ 27. $C=\begin{Vmatrix} \infty & 10 & 30 & 25 & 30 \\ 30 & \infty & 5 & 6 & 17 \\ 30 & 10 & \infty & 30 & 45 \\ 5 & 5 & 6 & \infty & 15 \\ 21 & 4 & 20 & 4 & \infty \end{Vmatrix}$

28. $C=\begin{Vmatrix} \infty & 5 & 9 & 8 & 6 & 8 \\ 3 & \infty & 8 & 10 & 13 & 11 \\ 4 & 7 & \infty & 7 & 6 & 9 \\ 4 & 10 & 6 & \infty & 6 & 11 \\ 11 & 12 & 3 & 3 & \infty & 3 \\ 16 & 13 & 10 & 6 & 6 & \infty \end{Vmatrix}$ 29. $C=\begin{Vmatrix} \infty & 17 & 37 & 47 & 37 \\ 37 & \infty & 13 & 20 & 28 \\ 35 & 15 & \infty & 41 & 51 \\ 14 & 14 & 12 & \infty & 30 \\ 48 & 37 & 44 & 20 & \infty \end{Vmatrix}$ 30. $C=\begin{Vmatrix} \infty & 15 & 25 & 25 & 5 \\ 25 & \infty & 6 & 2 & 15 \\ 2 & 3 & \infty & 5 & 12 \\ 15 & 14 & 16 & \infty & 2 \\ 4 & 15 & 3 & 15 & \infty \end{Vmatrix}$

2.4 Пошук максимального потоку транспортної мережі

Транспортною мережею (s/t-мережі) назвемо зв'язний орієнтований граф без петель, у якого є дві особливі вершини s (джерело) та t (стік). Причому з вершини s дуги тільки виходять, а у вершину t – тільки заходять; кожній дузі транспортної мережі приписана невід'ємна величина, яка називається пропускною спроможністю дуги і позначається c_{ij}.

Зафіксуємо деяку вершину s/t-мережі $x_к$. Для поняття потоку транспортної мережі розглянемо дві множини A-множина всіх вершин s/t-мережі, з яких виходить дуга та заходить у вершину $x_к$, та B – множина всіх вершин s/t-мережі, які безпосередньо йдуть за вершиною $x_к$. Тоді потоком у s/t-мережі будемо називати невід'ємну величину φ_{ij} (для деякої дуги u_{ij})=v, якщо виконуються такі умови:

$\varphi_{ij} \le c_{ij}$, для будь-якої дуги графа;

$$\sum_{i\in A}\varphi_{ik} - \sum_{j\in B}\varphi_{kj} = \begin{cases} -v, k = s; \\ 0\,, k \ne s \text{ та } \quad k \ne t; \\ v\,, \quad k = t. \end{cases}$$

Перша умова говорить, що для будь-якої дуги транспортної мережі потік не перевищує пропускну спроможність транспортної мережі, а друга про те, що для будь-якої вершини транспортної мережі сумарний потік, що входить у вершину, дорівнює сумарному потоку, що виходить з вершини, тобто потік неподільний.

Нехай X – множина всіх вершин s/t-мережі. Розіб'ємо її на дві підмножини X_1 та X_2 так, щоб:

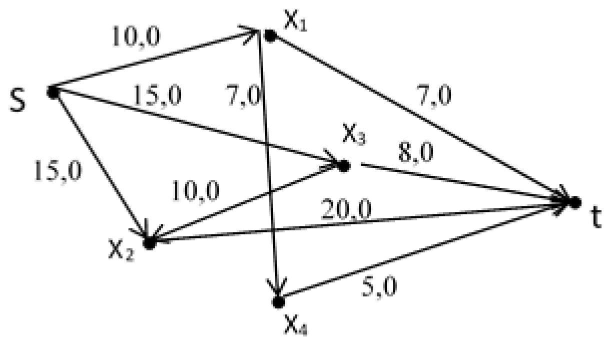
$$s \in X_1 \text{ та } s \notin X_2, a \;\; t \in X_2 \text{ та } t \notin X_1;$$
$$X_1 \bigcup X_2 = X, X_1 \bigcap X_2 = X.$$

Тоді розтином s/t-мережі назвемо множину всіх дуг, що йдуть з множини вершин X_1 до множини вершин X_2, позначимо (X_1, X_2). Величина розтину дорівнює сумі пропускних спроможностей дуг, що створюють розтин, та позначається $v(X_1, X_2)$.

Задача про визначення максимального потоку в транспортній мережі є основною задачею оптимізації на графах. Її розв'язання базується на теоремі Форда – Фалкерсона: максимальний потік у транспортній мережі дорівнює величині найменшого розтину в ній.

Розглянемо розв`язання подібної задачі на *прикладі 2.6*.

Транспортну мережу задано графічно. Необхідно знайти максимальний потік у s/t-мережі.

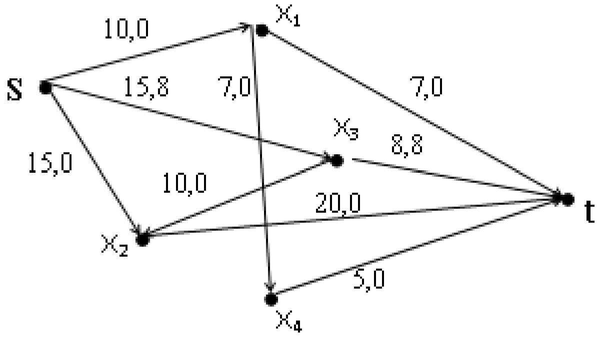


Розв'язання. Кожній дузі графа припишемо пару чисел c_{ij}, φ_{ij}. На початку в s/t-мережі розподілимо нульовий потік, тобто $\varphi_{ij}=0$. Далі для розмітки будь-якої к-ї вершини будемо використовувати позначення $к(i^+, \varepsilon(к))$ або $к(i^-, \varepsilon(к))$, де i-номер вершини, з якої ми потрапили до вершини к, '+' – по прямій дузі, '-' – по зворотній дузі, $\varepsilon(к)$ – невід'ємна величина, на яку можна збільшити потік (для прямої дуги) або зменшити (для зворотної дуги). Вершина s завжди має особливу помітку (s^+, ∞). Помічати вершини по прямих дугах можна, якщо на цих дугах потік можливо збільшити ($c_{ij}-\varphi_{ij}>0$), а по зворотніх – якщо потік можливо зменшити ($\varphi_{ij}>0$). Для вершини $к(i^+, \varepsilon(к))$ $\varepsilon(к)=\min(\varepsilon(i), c_{ik}-\varphi_{ik})$, а для вершини $к(i^-, \varepsilon(к))$ $\varepsilon(к)=\min(\varepsilon(i), \varphi_{ik})$. Перший крок будь-якого етапу полягає у розставленні поміток вершин мережі, доки або не буде помічено вершину t , тоді переходимо на другий крок, або переконаємося, що вершину t помітити неможливо (задачу розв'язано).

1-й етап. 1-й крок. $S(s^+,\infty)$. Далі помічаємо вершини, суміжні з s.

$1(s^+, \min(\infty,10\text{-}0)$, тобто $1(s^+,10)$.

Аналогічно $2(s^+,15)$, $3(s^+,15)$. Тепер вершина s розглянута. Для подальшого розгляду вибираємо будь-яку із суміжних помічених вершин. Наприклад, вершину 3. Помічаємо вершини, суміжні з третьою: $2(3^+, \min(15, 10\text{-}0)=10)$, $t(3^+, \min(15,8\text{-}0)=8)$. Помітили вершину t – переходимо до наступного кроку 1-го етапу.



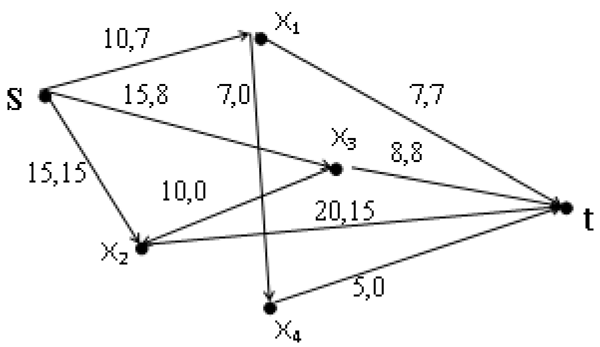
2-й крок. За допомогою поміток вершин будуємо ланцюг, який приводить з вершини s до вершини t: $t \leftarrow 3 \leftarrow s$. На прямих дугах ланцюга потік збільшуємо на величину $\varepsilon(t)$, а на зворотній – зменшуємо на цю саму величину. На рисунку показано розподіл потоку після виконання першого етапу. Переходимо до другого етапу, який виконуємо аналогічно першому.

2-й етап

$S(s^+,\infty)$	$1(s^+,10)$ $2(s^+,15)$ $3(s^+,7)$	$t(2^+,15)$ $t\leftarrow 2\leftarrow s$ $\varepsilon(t)=15$

3-й етап

$S(s^+,\infty)$	$1(s^+,10)$ $3(s^+,7)$	$4(1^+,7)$ $t(1^+,7)$ $t\leftarrow 1\leftarrow s$ $\varepsilon(t)=7$



Після 2-го та 3-го етапів розподіл потоку в мережі показано на рисунку.

4-й етап

$S(s^+,\infty)$	$1(s^+,3)$ $3(s^+,7)$	$2(3^+,7)$	$t(2^+,5)$ $t\leftarrow 2\leftarrow 3\leftarrow s$ $\varepsilon(t)=5$

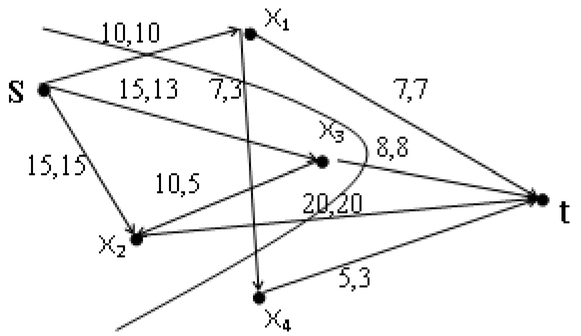
5-й етап.

$S(s^+,\infty)$	$1(s^+,3)$ $3(s^+,2)$	$4(1^+,3)$	$t(4^+,3)$ $t\leftarrow 4\leftarrow 1\leftarrow s$ $\varepsilon(t)=3$

6-й етап.

$S(s^+,\infty)$	$3(s^+,2)$	$2(3^+,2)$	$S(2^-,2)$

На 1-му кроці 6-го етапу прийшли до побудови контуру навколо вершини s, а до вершини t так і не потрапили. Це свідчить про те, що потік у мережі збільшити неможливо. Будуємо розтин мережі: віднесемо до множини X_1 усі вершини, що помічені на 6-му кроці, а до множини X_2 – ті, що не помічені. $X_1=\{s, 2, 3\}$, $X_2=\{1, 4, t\}$, $(X_1, X_2)=\{u_{s1}, u_{2t}, u_{3t}\}$, $v(X_1, X_2)=c_{s1}+c_{2t}+c_{3t}=10+20+8=38$. Перевірка: якщо просумувати всі значення $\varepsilon(t)$: 8+15+7+5+3=38, то це число збігається з величиною побудованого розтину.



Візьмемо вихідний граф і лінією відокремимо множину вершин X_1 від множини вершин X_2. Ті дуги, які перетинають отриману лінію і мають напрямок від вершин з множини X_1 до вершин з множини X_2, і є дугами розтину.

На рисунку також показано результуючий розподіл потоку в мережі.

Завдання № 6

Транспортна мережа задана за допомогою таблиці, де зазначено дуги мережі та пропускну спроможність цих дуг. Необхідно зобразити граф графічно і, використовуючи теорему Форда-Фалкерсона, визначити найбільший потік у мережі (s – джерело, t – стік).

№	Перелік дуг мережі у вигляді u_{ij} (*дуга* $i \to j$) $= c_{ij}$ (*пропускна спроможність*)
1	$u_{s1}=10$, $u_{s2}=7$, $u_{s3}=6$, $u_{1t}=4$, $u_{14}=2$, $u_{13}=1$, $u_{23}=2$, $u_{21}=5$, $u_{34}=3$, $u_{3t}=3$, $u_{4t}=12$
2	$u_{s1}=16$, $u_{s2}=20$, $u_{s3}=4$, $u_{1t}=10$, $u_{13}=8$, $u_{23}=5$, $u_{24}=20$, $u_{34}=4$, $u_{3t}=5$, $u_{4t}=15$
3	$u_{s1}=4$, $u_{s2}=2$, $u_{s3}=5$, $u_{s4}=7$, $u_{13}=1$, $u_{14}=2$, $u_{23}=1$, $u_{24}=1$, $u_{3t}=9$, $u_{43}=3$, $u_{4t}=5$
4	$u_{s1}=6$, $u_{s2}=5$, $u_{s4}=14$, $u_{1t}=5$, $u_{12}=3$, $u_{23}=8$, $u_{3t}=10$, $u_{43}=4$, $u_{45}=8$, $u_{53}=5$, $u_{5t}=7$
5	$u_{s1}=6$, $u_{s2}=14$, $u_{1t}=10$, $u_{14}=3$, $u_{23}=3$, $u_{21}=10$, $u_{24}=2$, $u_{34}=5$, $u_{4t}=8$
6	$u_{s1}=16$, $u_{s2}=14$, $u_{13}=9$, $u_{21}=10$, $u_{23}=5$, $u_{24}=8$, $u_{35}=6$, $u_{3t}=9$, $u_{45}=1$, $u_{4t}=6$, $u_{5t}=6$
7	$u_{s1}=6$, $u_{s2}=10$, $u_{s3}=3$, $u_{12}=3$, $u_{14}=2$, $u_{13}=2$, $u_{24}=1$, $u_{2t}=8$, $u_{32}=4$, $u_{34}=5$, $u_{4t}=9$
8	$u_{s1}=3$, $u_{s2}=5$, $u_{s3}=4$, $u_{s4}=1$, $u_{1t}=7$, $u_{21}=2$, $u_{24}=1$, $u_{2t}=1$, $u_{32}=3$, $u_{34}=2$, $u_{4t}=3$
9	$u_{s1}=25$, $u_{s2}=18$, $u_{12}=4$, $u_{14}=10$, $u_{24}=10$, $u_{2t}=15$, $u_{3t}=5$, $u_{43}=10$, $u_{4t}=10$
10	$u_{s1}=5$, $u_{s2}=4$, $u_{s3}=6$, $u_{1t}=4$, $u_{14}=1$, $u_{13}=1$, $u_{23}=3$, $u_{24}=2$, $u_{34}=2$, $u_{3t}=3$, $u_{4t}=7$
11	$u_{s1}=12$, $u_{s2}=20$, $u_{s3}=5$, $u_{1t}=10$, $u_{13}=8$, $u_{23}=5$, $u_{24}=10$, $u_{34}=4$, $u_{3t}=8$, $u_{4t}=12$
12	$u_{s1}=8$, $u_{s3}=4$, $u_{s4}=6$, $u_{12}=4$, $u_{24}=2$, $u_{2t}=10$, $u_{31}=3$, $u_{32}=2$, $u_{34}=5$, $u_{4t}=7$
13	$u_{s1}=10$, $u_{s2}=10$, $u_{s3}=15$, $u_{14}=7$, $u_{12}=4$, $u_{24}=6$, $u_{32}=5$, $u_{3t}=20$, $u_{4t}=10$
14	$u_{s1}=6$, $u_{s2}=7$, $u_{s3}=5$, $u_{12}=3$, $u_{14}=2$, $u_{13}=1$, $u_{23}=4$, $u_{24}=2$, $u_{2t}=3$, $u_{3t}=7$, $u_{4t}=8$
15	$u_{s1}=5$, $u_{s2}=6$, $u_{s3}=8$, $u_{12}=2$, $u_{14}=3$, $u_{13}=1$, $u_{24}=1$, $u_{2t}=4$, $u_{3t}=5$, $u_{4t}=6$

№	Перелік дуг мережі у вигляді u_{ij} (*дуга* $i \to j$) $= c_{ij}$ (*пропускна спроможність*)
16	u_{s1}=10, u_{s2}=3, u_{s3}=15, u_{1t}=5, u_{12}=8, u_{23}=16, u_{24}=4, u_{2t}=15, u_{34}=5, u_{4t}=20
17	u_{s1}=7, u_{s4}=15, u_{st}=5, u_{1t}=3, u_{13}=5, u_{21}=10, u_{34}=3, u_{42}=20, u_{4t}=6
18	u_{s1}=20, u_{s4}=5, u_{12}=17, u_{13}=6, u_{24}=10, u_{2t}=3, u_{32}=10, u_{3t}=5, u_{4t}=20
19	u_{s1}=25, u_{s3}=40, u_{14}=20, u_{13}=15, u_{2t}=40, u_{32}=20, u_{3t}=15, u_{4t}=7
20	u_{s1}=5, u_{s2}=18, u_{s5}=7, u_{14}=15, u_{23}=25, u_{35}=10, u_{3t}=4, u_{43}=10, u_{4t}=10, u_{5t}=2
21	u_{s1}=20, u_{s2}=10, u_{s3}=15, u_{12}=15, u_{23}=20, u_{25}=15,u_{34}=4,u_{35}=8, u_{3t}=1,u_{4t}=20,u_{5t}=4
22	u_{s1}=10, u_{s2}=5, u_{s4}=10, u_{12}=1, u_{13}=5, u_{14}=8, u_{34}=4, u_{3t}=4, u_{42}=5, u_{2t}=20
23	u_{s1}=15, u_{s2}=20, u_{s3}=5, u_{1t}=15, u_{12}=8, u_{23}=16, u_{24}=4, u_{2t}=10, u_{34}=5, u_{4t}=3
24	u_{s1}=10, u_{s3}=15, u_{14}=8, u_{1t}=13, u_{24}=15, u_{31}=6, u_{32}=3, u_{3t}=5, u_{4t}=12
25	u_{s3}=10, u_{s4}=15, u_{1t}=13, u_{2t}=20, u_{31}=8, u_{34}=5, u_{41}=2, u_{42}=8, u_{4t}=12
26	u_{s1}=15, u_{s2}=12, u_{s3}=11, u_{1t}=9, u_{14}=7, u_{13}=6, u_{23}=7, u_{21}=10, u_{34}=8, u_{3t}=8, u_{4t}=17
27	u_{s1}=13, u_{s2}=17, u_{s3}=1, u_{1t}=7, u_{13}=5, u_{23}=2, u_{24}=17, u_{34}=1, u_{3t}=2, u_{4t}=12
28	u_{s1}=9, u_{s2}=7, u_{s3}=10, u_{s4}=12, u_{13}=6, u_{14}=8, u_{23}=6, u_{24}=6, u_{3t}=14, u_{43}=8, u_{4t}=10
29	u_{s1}=8, u_{s2}=7, u_{s4}=16, u_{1t}=7, u_{12}=5, u_{23}=10, u_{3t}=12, u_{43}=6, u_{45}=10, u_{53}=7, u_{5t}=9
30	u_{s1}=11, u_{s2}=19, u_{1t}=15, u_{14}=8, u_{23}=8, u_{21}=15, u_{24}=7, u_{34}=10, u_{4t}=13, u_{2t}=2

2.5 Пошук остовного дерева мінімальної ваги. Алгоритми Краскала та Пріма.

Остовне (каркасне, стяжне, кістякове) дерево неорієнтовного графа – це такий підграф, що: 1) множина вершин під графа співпадає з множиною вершин початкового графа; 2) підграф є деревом.

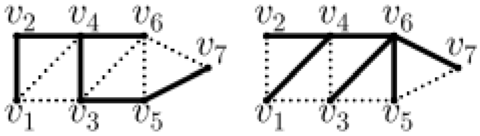


Рисунок 2.5.1. – Приклад остовних дерев

Кількість ребер будь-якого остовного дерева дорівнює n-1. Але для зважених графів сума ребер остовних дерев може біти різною. Тому виникає задача побудови остовного дерева мінімальної ваги: серед усіх можливих дерев знайти те, сума довжин ребер якого якнайменша.

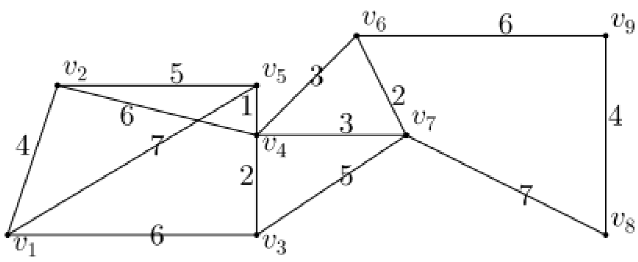
Приклад необхідності побудови мінімального остовного дерева: нехай є сукупність міст, між якими треба прокласти залізницю; для кожної пари міст відома оцінка (наприклад кошторис) технічної можливості прокладання залізниці. Якщо поставити за мету побудувати залізницю так, щоб з одного міста можна було потрапити до будь-якого іншого (можливо, з пересадками), і сумарна вартість будівництва була як найменшою, то в результаті отримаємо задачу побудови найкоротшого остовного дерева.

Зазвичай для розв'язку цієї задачі застосовують алгоритми Пріма або Краскала. Також популярним є об'єднання цих двох методів у алгоритм Пріма-Краскала.

Алгоритм Краскала. Усі ребра графа необхідно розташувати в неспадному порядку за значенням ваги ребра. Спочатку розглядається вихідний граф у вигляді лісу (Ліс – це граф, у якого немає циклів і для якого поняття зв'язності є необов'язковим. Очевидно, ліс складається з дерев, які є його компонентами зв'язності.), який містить усі вершини початкового графа і не містить жодного ребра, це дозволяє розглянути кожну вершину графа як окрему компоненту зв'язності.

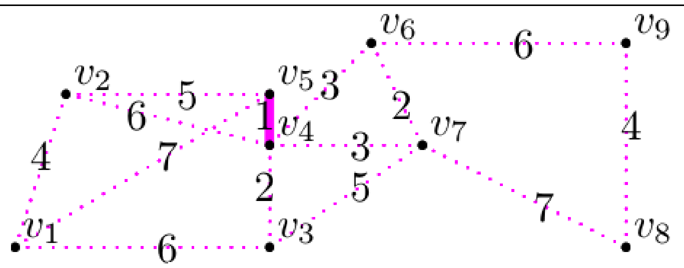
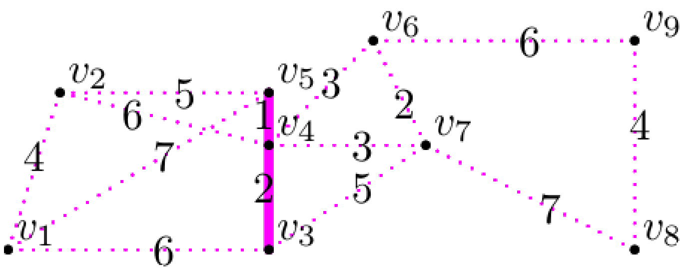
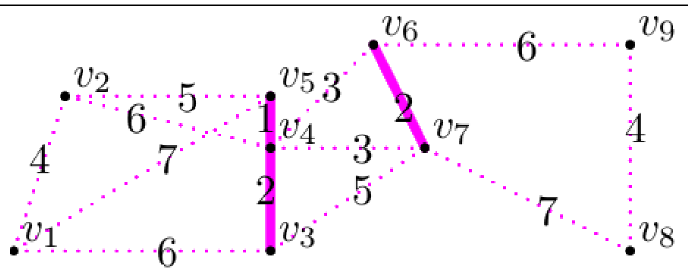
Далі з масиву відсортованих ребер необхідно почергово переглядати ребра від найменшого до найбільшого і включати до дерева лише ті ребра, вершини якого належать до різних компонент зв'язності графа. Якщо ребро має вершини, що належать до однієї компоненти зв'язності графа, то його включення до графа призвело б до появи одного елементарного циклу. Процес включення ребер до графа продовжуємо доти, поки не буде створено остовне дерево.

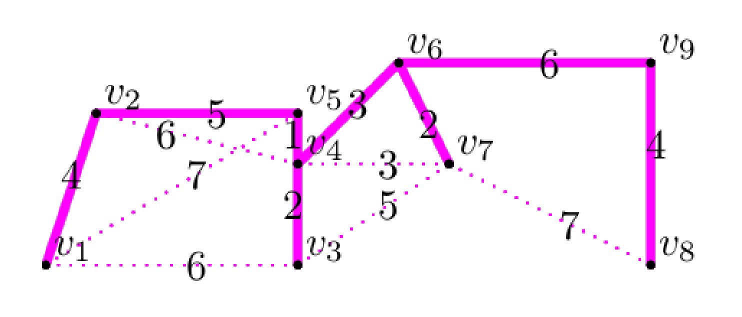
Приклад побудови остовного дерева за алгоритмом Краскала: задано граф графічно



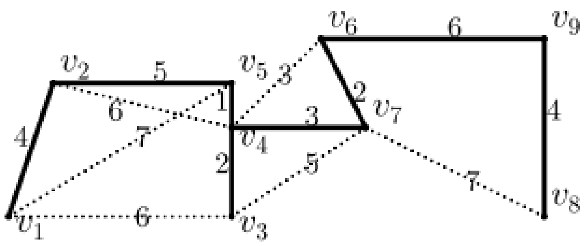
Випишемо усі ребра графа і відсортуємо їх за вагою у неспадному порядку: ({v_4 , v_5 }, 1), ({v_3 , v_4 }, 2), ({v_6 , v_7 }, 2), ({v_4 , v_6 }, 3), ({v_4 , v_7 }, 3), ({v_1 , v_2 }, 4), ({v_8 , v_9 }, 4), ({v_2 , v_5 }, 5), ({v_3 , v_7 }, 5) , ({v_1 , v_3 }, 6), ({v_2 , v_4 }, 6), ({v_6 , v_9 }, 6), ({v_1 , v_5 }, 7), ({v_7 , v_8 }, 7).

Далі починаємо перебирати ребра, намагаючись кожне ребро включити до дерева:

граф	Опис дії
	Розглядаємо ребро ({v_4 , v_5 }, 1). Його кінці v_4 та v_5 належать до різних компонент зв'язності, тому додаємо це ребро
	Розглядаємо ребро ({v_3 , v_4 }, 2). Його кінці v_3 та v_4 належать до різних компонент зв'язності, тому додаємо це ребро
	Розглядаємо ребро ({v_6 , v_7 }, 2). Його кінці v_6 та v_7 належать різним компонентам зв'язності, тому додаємо дане ребро
…	…

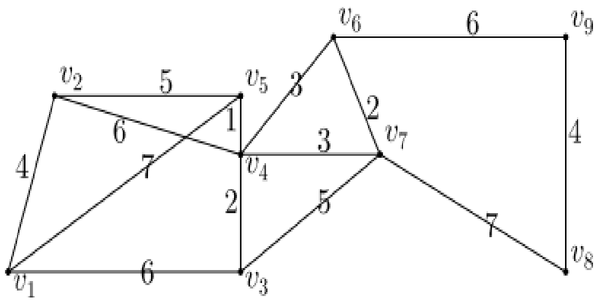
	Так продовжуємо переглядати послідовно усі ребра з відсортованого списку аж до останнього вдалого включення ребра ($\{v_6, v_9\}$, 6)

Два останні ребра зі списку не будуть включені. Сумарна довжина побудованого дерева дорівнює 27. Але для даного графа існує ще одне мінімальне остовне дерево довжиною 27, яке можна отримати, обравши під час сортування ребро ($\{v_4, v_7\}$, 3) взамін ребра ($\{v_4, v_6\}$, 3):



Алгоритм Пріма. Як і в алгоритмі Краскала, початковий граф розглядається у вигляді лісу вершин, після чого виконуються кроки додавання ребер. Але в алгоритмі Пріма ребра додаються так, щоб вже побудована частина постійно була деревом. Спочатку вибирається довільна вершина графа, як початкове дерево. Потім, серед ребер, один кінець яких належить дереву, а інший не належить, обирають найкоротше і включать до дерева. Цей процес продовжується доти, поки не будуть задіяні усі вершини.

Приклад побудови остовного дерева за алгоритмом Пріма розглянемо для графа, який використовувався як початковий для алгоритму Краскала. Для зручності використання методу краще подати вихідний граф у вигляді матриці суміжності вершин.



$$C=\begin{pmatrix}\infty & 4 & 6 & \infty & 7 & \infty & \infty & \infty & \infty\\ 4 & \infty & \infty & 6 & 5 & \infty & \infty & \infty & \infty\\ 6 & \infty & \infty & 2 & \infty & \infty & 5 & \infty & \infty\\ \infty & 6 & 2 & \infty & 1 & 3 & 3 & \infty & \infty\\ 7 & 5 & \infty & 1 & \infty & \infty & \infty & \infty & \infty\\ \infty & \infty & \infty & 3 & \infty & \infty & 2 & \infty & 6\\ \infty & \infty & 5 & 3 & \infty & 2 & \infty & 7 & \infty\\ \infty & \infty & \infty & \infty & \infty & \infty & 7 & \infty & 4\\ \infty & \infty & \infty & \infty & \infty & 6 & \infty & 4 & \infty\end{pmatrix}$$

Як початкову вершину дерева оберемо v_1. Для зручності сформуємо вектор d з ваг суміжних ребер дерева побудови. Оскільки на початку остовне дерево – це лише одна вершина, то початкові значення вектора d будуть відповідати першому рядку матриці суміжності ваг графа. У векторі d будемо ставити символ «-» проти вершини, яка вже була включена до остовного дерева (на початковому етапі це вершина v_1). Результати перерахунку елементів вектора d :

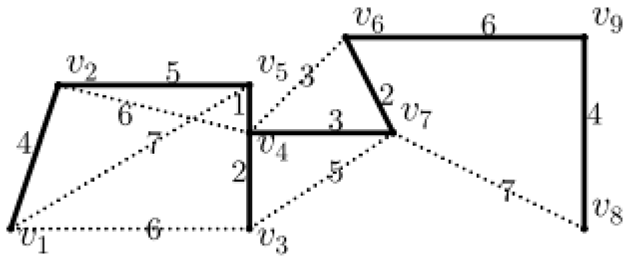
№	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	Обране ребро
1	-	**4**	6	∝	7	∝	∝	∝	∝	$(\{v_1, v_2\}, 4)$

Отже, за перший крок алгоритму до остовного дерева було включене одне ребро або додана вершина v_2. При формуванні подальших значень елементів вектора d будемо порівнювати значення елементів вектору з відповідними рядком матриці суміжності ваг графа до обраної вершини . В нашому випадку це елементи 2-го рядка (тому, що обрана вершина v2). Серед елементів, що порівнюються, слід обирати найменше із значень.

№	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	Обране ребро
2	-	-	6	6	**5**	∝	∝	∝	∝	$(\{v_2, v_5\}, 5)$
3	-	-	6	**1**	-	∝	∝	∝	∝	$(\{v_4, v_5\}, 1)$

4	-	-	**2**	-	-	3	3	$\propto$	$\propto$	({v_3 , v_4 }, 2)
5	-	-	-	-	-	3	**3**	$\propto$	$\propto$	({v_4 , v_7 }, 3)
6	-	-	-	-	-	**2**	-	7	$\propto$	({v_6 , v_7 }, 2)
7	-	-	-	-	-	-	-	7	**6**	({v_6 , v_9 }, 6)
8	-	-	-	-	-	-	-	**4**	-	({v_8 , v_9 }, 4)

З обраних ребер отримано одне з двох мінімальних остовних дерев сумарною вагою ребер 27, яке співпадає з результатом побудови остовного дерева за алгоритмом Краскала.



Завдання № 7

Неорієнтований зв’язний граф задано переліком ребер {x_i, x_j} та відповідно вагою c_{ij} кожного ребра ({x_i, x_j},c_{ij}). Необхідно знайти мінімальні остовні дерева заданого графа за алгоритмами Краскала та Пріма. Отримані дерева зобразити графічно.

Варіанти опису графів.

1. ({1,2},5), ({1,4},12), ({1,5},3),({2,3},11), ({2,5},8), ({2,6},18), ({3,4},12), ({3,5},2), ({3,6},11), ({3,7},10), ({3,8},3), ({4,5},5), ({4,7},18), ({4,8},21), ({4,11},2), ({5,7}, 8), ({6,9},3), ({6,11},2), ({7,8},2), ({7,9},15), ({7,10},11), ({8,10},1), ({8,11},8), ({9,10},12).
2. ({1,5},8), ({1,9},7), ({1,13},6), ({2,3},5), ({2,4},10), ({2,5},1), ({2,13},12), ({3,4},15), ({3,7},21), ({3,15},8), ({3,18),1), ({4,9},3), ({4,13), 7), ({4,15},4), ({5,13},3), ({5,15},10), ({5,18},15), ({7,8},2), ({7,15},3), ({8,15},18), ({8,18},8), ({9,13},12), ({15,18},15).

3. ({a,b},18), ({a,c},5), ({a,k},3), (a,v},2), ({b,c},10), ({b,i},9), ({b,s},2), ({c,d},11), ({c,s},40), ({c,v},2), ({d,e},8), ({d,j},2), ({d,s},22), ({d,v},15), ({e,j},8), ({e,s},4), ({h,i},3), ({h,j},20), ({h,s},18), ({j,s},18), ({i,s},2), ({i,v},14), ({k,v},12), ({s,v},3).
4. ({1,2},5), ({1,10},2), ({1,9},21), (2,3},2), ({2,4},15), ({2,8},8), ({2,10},18), ({3,5),3), ({3,6},1), ({3,9},13), (3,10},5), ({4,6},8), ({4,8},4), ({4,5},11), ({4,10},15), ({5,7},8), ({5,8},2), ({5,10},7), ({6,9},5), ({6,11},2), ({7,8},3), ({7,9},12), ({8,11},18), ({9,10},16).
5. ({1,2},20), ({1,6},8),({1,10},12), ({1,13},2), ({2,5},3), (2,6},3), ({2,7},6), ({2,13},10), ({3,4},7),({3,5},4), ({3,7},2), ({3,11},13), ({4,5},10), ({4,8},5), ({4,9},8), ({4,13},8), ({5,9},15),({5,13},7), ({6,13},15), ({6,8},3), ({8,9},4), ({8,11},3), ({8,13},10), ({9,11},6).
6. ({1,2},3), ({1,3},14), ({1,4},6), ({1,5},2), ({1,8},4), ({1,9},10), ({2,3},8), ({2,5},2), ({2,11},3), ({3,7},5), ({3,9},6), ({4,6},5), ({4,9},1), ({4,10},3), ({4,11},15), ({5,10},16), ({5,11},12), ({6,7},5), ({6,8},8), ({6,9},2), ({6,11},3), ({7,8),7), ({8,9},13), ({9,10},6), ({9,11},18), ({10,11},12).
7. ({1,4},11), ({1,6},8), ({1,12},6), ({2,4},5), ({2,5},3), ({2,15},12), ({3,5},6), ({3,6},1), ({3,7},15), ({3,8},,18), ({3,10},6), ({3,12},9), ({3,66},8), ({4,5},6), ({4,6},4), ({5,6},3), ({5,8},7), ({5,15},11), ({6,8},4), ({6,7},13), ({6,12},2), ({7,10},2), ({7,12}.3), ({7,66},6), ({8,10},5), ({8,15},2), ({8,66},7), ({10,15},3).
8. ({1,3},5), ({1,4},6), ({1,7},12), ({1,8},5), ({1,14},2), ({3,5},13), ({3,7},7), ({3,8}, 6), ({3,9},2), ({3,14},14), ({4,5},11), ({4,16},18), ({5,9},4), ({5,16},6), ({6,7},8), ({6,8},3), ({6,14},3), ({6,15},17), ({7,8},13), ({7,9},15), ({7,15},9), ({8,9},4),({8,15},11), ({8,16},2), ({9,16},21), ({9,14},4), ({14,15},2).
9. ({1,2},3), ({1,3},2), ({1,4},3), ({1,5},8), ({1,7},4), ({1,11},13), ({2,3},15), ({2,5},6), ({2,7},8), ({2,12},14), ({3,4},4), ({3,5},10), ({3,12},12), ({3,21},7), ({3,27},6), ({4,7},2), ({4,11),2), ({4,21},5), (5,12},3), (5,21},12), ({7,44},10),

({11,21},4), ({11,27},18), ({12,21},16), ({12,44},10), ({21,27},9), ({27,44},11).

10.({1,5},15), ({1,6},14), ({1,7},10), ({1,9},3), ({1,10},7), ({2,3},10), ({2,8},11), ({2,11},7), ({2,10},6), ({3,4},1), ({3,10},5), ({3,11},2), ({4,5},10), ({4,10},12), ({5,8},10), ({5,9},7), ({6,7},18), ({6,8},8), ({6,11},2), ({8,9},6), ({10,11},12).

11.({1,2},5), ({1,4},12), ({1,5},3),({2,3},11), ({2,5},8), ({2,6},18), ({3,4},12), ({3,5},2), ({3,6},11), ({3,7},10), ({3,8},3), ({4,5},5), ({4,7},18), ({4,8},21), ({4,11},2), ({5,7}, 8), ({6,9},3), ({6,11},2), ({7,8},2), ({7,9},15), ({7,10},11), ({8,10},1), ({8,11},8), ({9,10},12).

12.({1,5},8), ({1,9},7), ({1,13},6), ({2,3},5), ({2,4},10), ({2,5},1), ({2,13},12), ({3,4},15), ({3,7},21), ({3,15},8), ({3,18),1), ({4,9},3), ({4,13), 7), ({4,15},4), ({5,13},3), ({5,15},10), ({5,18},15), ({7,8},2), ({7,15},3), ({8,15},18), ({8,18},8), ({9,13},12), ({15,18},15).

13.({a,b},18), ({a,c},5), ({a,k},3), (a,v},2), ({b,c},10), ({b,i},9), ({b,s},2), ({c,d},11), ({c,s},40), ({c,v},2), ({d,e},8), ({d,j},2), ({d,s},22), ({d,v},15), ({e,j},8), ({e,s},4), ({h,i},3), ({h,j},20), ({h,s},18), ({j,s},18), ({i,s},2), ({i,v},14), ({k,v},12), ({s,v},3).

14.({1,2},5), ({1,10},2), ({1,9},21), (2,3},2), ({2,4},15), ({2,8},8), ({2,10},18), ({3,5),3), ({3,6},1), ({3,9},13), (3,10},5), ({4,6},8), ({4,8},4), ({4,5},11), ({4,10},15), ({5,7},8), ({5,8},2), ({5,10},7), ({6,9},5), ({6,11},2), ({7,8},3), ({7,9},12), ({8,11},18), ({9,10},16).

15.({1,2},20), ({1,6},8),({1,10},12), ({1,13},2), ({2,5},3), (2,6},3), ({2,7},6), ({2,13},10), ({3,4},7),({3,5},4), ({3,7},2), ({3,11},13), ({4,5},10), ({4,8},5), ({4,9},8), ({4,13},8), ({5,9},15),({5,13},7), ({6,13},15), ({6,8},3), ({8,9},4), ({8,11},3), ({8,13},10), ({9,11},6).

16.({1,2},3), ({1,3},14), ({1,4},6), ({1,5},2), ({1,8},4), ({1,9},10), ({2,3},8), ({2,5},2), ({2,11},3), ({3,7},5), ({3,9},6), ({4,6},5), ({4,9},1), ({4,10},3), ({4,11},15), ({5,10},16), ({5,11},12), ({6,7},5), ({6,8},8), ({6,9},2), ({6,11},3), ({7,8),7), ({8,9},13), ({9,10},6), ({9,11},18), ({10,11},12).

17.({1,4},11), ({1,6},8), ({1,12},6), ({2,4},5), ({2,5},3), ({2,15},12), ({3,5},6), ({3,6},1), ({3,7},15), ({3,8},,18), ({3,10},6), ({3,12},9), ({3,66},8), ({4,5},6), ({4,6},4), ({5,6},3), ({5,8},7), ({5,15},11), ({6,8},4), ({6,7},13), ({6,12},2), ({7,10},2), ({7,12}.3), ({7,66},6), ({8,10},5), ({8,15},2), ({8,66},7), ({10,15},3).

18.({1,3},5), ({1,4},6), ({1,7},12), ({1,8},5), ({1,14},2), ({3,5},13), ({3,7},7), ({3,8}, 6), ({3,9},2), ({3,14},14), ({4,5},11), ({4,16},18), ({5,9},4), ({5,16},6), ({6,7},8), ({6,8},3), ({6,14},3), ({6,15},17), ({7,8},13), ({7,9},15), ({7,15},9), ({8,9},4),({8,15},11), ({8,16},2), ({9,16},21), ({9,14},4), ({14,15},2).

19.({1,2},3), ({1,3},2), ({1,4},3), ({1,5},8), ({1,7},4), ({1,11},13), ({2,3},15), ({2,5},6), ({2,7},8), ({2,12},14), ({3,4},4), ({3,5},10), ({3,12},12), ({3,21},7), ({3,27},6), ({4,7},2), ({4,11),2), ({4,21},5), (5,12},3), (5,21},12), ({7,44},10), ({11,21},4), ({11,27},18), ({12,21},16), ({12,44},10), ({21,27},9), ({27,44},11).

20.({1,5},15), ({1,6},14), ({1,7},10), ({1,9},3), ({1,10},7), ({2,3},10), ({2,8},11), ({2,11},7), ({2,10},6), ({3,4},1), ({3,10},5), ({3,11},2), ({4,5},10), ({4,10},12), ({5,8},10), ({5,9},7), ({6,7},18), ({6,8},8), ({6,11},2), ({8,9},6), ({10,11},12).

21.({1,2},5), ({1,4},12), ({1,5},3),({2,3},11), ({2,5},8), ({2,6},18), ({3,4},12), ({3,5},2), ({3,6},11), ({3,7},10), ({3,8},3), ({4,5},5), ({4,7},18), ({4,8},21), ({4,11},2), ({5,7}, 8), ({6,9},3), ({6,11},2), ({7,8},2), ({7,9},15), ({7,10},11), ({8,10},1), ({8,11},8), ({9,10},12).

22.({1,5},8), ({1,9},7), ({1,13},6), ({2,3},5), ({2,4},10), ({2,5},1), ({2,13},12), ({3,4},15), ({3,7},21), ({3,15},8), ({3,18),1), ({4,9},3), ({4,13), 7), ({4,15},4), ({5,13},3), ({5,15},10), ({5,18},15), ({7,8},2), ({7,15},3), ({8,15},18), ({8,18},8), ({9,13},12), ({15,18},15).

23.({a,b},18), ({a,c},5), ({a,k},3), (a,v},2), ({b,c},10), ({b,i},9), ({b,s},2), ({c,d},11), ({c,s},40), ({c,v},2), ({d,e},8), ({d,j},2), ({d,s},22), ({d,v},15), ({e,j},8), ({e,s},4), ({h,i},3), ({h,j},20), ({h,s},18), ({j,s},18), ({i,s},2), ({i,v},14), ({k,v},12), ({s,v},3).

24.({1,2},5), ({1,10},2), ({1,9},21), (2,3},2), ({2,4},15), ({2,8},8), ({2,10},18), ({3,5),3), ({3,6},1), ({3,9},13), (3,10},5), ({4,6},8), ({4,8},4), ({4,5},11), ({4,10},15), ({5,7},8), ({5,8},2), ({5,10},7), ({6,9},5), ({6,11},2), ({7,8},3), ({7,9},12), ({8,11},18), ({9,10},16).

25.({1,2},20), ({1,6},8),({1,10},12), ({1,13},2), ({2,5},3), (2,6},3), ({2,7},6), ({2,13},10), ({3,4},7),({3,5},4), ({3,7},2), ({3,11},13), ({4,5},10), ({4,8},5), ({4,9},8), ({4,13},8), ({5,9},15),({5,13},7), ({6,13},15), ({6,8},3), ({8,9},4), ({8,11},3), ({8,13},10), ({9,11},6).

26.({1,2},3), ({1,3},14), ({1,4},6), ({1,5},2), ({1,8},4), ({1,9},10), ({2,3},8), ({2,5},2), ({2,11},3), ({3,7},5), ({3,9},6), ({4,6},5), ({4,9},1), ({4,10},3), ({4,11},15), ({5,10},16), ({5,11},12), ({6,7},5), ({6,8},8), ({6,9},2), ({6,11},3), ({7,8),7), ({8,9},13), ({9,10},6), ({9,11},18), ({10,11},12).

27.({1,4},11), ({1,6},8), ({1,12},6), ({2,4},5), ({2,5},3), ({2,15},12), ({3,5},6), ({3,6},1), ({3,7},15), ({3,8},,18), ({3,10},6), ({3,12},9), ({3,66},8), ({4,5},6), ({4,6},4), ({5,6},3), ({5,8},7), ({5,15},11), ({6,8},4), ({6,7},13), ({6,12},2), ({7,10},2), ({7,12}.3), ({7,66},6), ({8,10},5), ({8,15},2), ({8,66},7), ({10,15},3).

28.({1,3},5), ({1,4},6), ({1,7},12), ({1,8},5), ({1,14},2), ({3,5},13), ({3,7},7), ({3,8}, 6), ({3,9},2), ({3,14},14), ({4,5},11), ({4,16},18), ({5,9},4), ({5,16},6), ({6,7},8), ({6,8},3), ({6,14},3), ({6,15},17), ({7,8},13), ({7,9},15), ({7,15},9), ({8,9},4),({8,15},11), ({8,16},2), ({9,16},21), ({9,14},4), ({14,15},2).

29.({1,2},3), ({1,3},2), ({1,4},3), ({1,5},8), ({1,7},4), ({1,11},13), ({2,3},15), ({2,5},6), ({2,7},8), ({2,12},14), ({3,4},4), ({3,5},10), ({3,12},12), ({3,21},7), ({3,27},6), ({4,7},2), ({4,11),2), ({4,21},5), (5,12},3), (5,21},12), ({7,44},10), ({11,21},4), ({11,27},18), ({12,21},16), ({12,44},10), ({21,27},9), ({27,44},11).

30.({1,5},15), ({1,6},14), ({1,7},10), ({1,9},3), ({1,10},7), ({2,3},10), ({2,8},11), ({2,11},7), ({2,10},6), ({3,4},1), ({3,10},5), ({3,11},2), ({4,5},10), ({4,10},12), ({5,8},10), ({5,9},7), ({6,7},18), ({6,8},8), ({6,11},2), ({8,9},6), ({10,11},12).

КРИТЕРІЙ ОЦІНЮВАННЯ

Відповідно до робочої програми навчальної дисципліни «Дискретна математика», на виконання індивідуального науково-дослідного завдання, яким слід уважати розрахунково-графічну роботу (РГР), відводиться 50 год (зі 150 год на дисципліну у семестрі), що становить половину (усього 100 год) часу, який відведено студенту на самостійну роботу з дисципліни.

Оскільки з відповідної дисципліни у семестрі передбачено як підсумковий контроль екзамен, то за поточну роботу в семестрі студент може отримати максимально 80 балів.

За виконання, оформлення та захист РГР студент може отримати 28 балів. Тобто кожне завдання максимально може бути оцінене у 4 бали. Це складова оцінка, що поєднує в собі:

2 бали за правильне виконання завдання;

1 бал за відповідне оформлення завдання;

1 бал за своєчасний захист завдання, на якому студент у спілкуванні з викладачем відповідає на теоретичні та практичні запитання відповідно до тематики завдання.

Оскільки семестр поділено на два змістових модулі, то написання першої модульної роботи необхідно виконати, оформити та захистити перші 4 завдання, а до написання другої модульної роботи – останні 3 завдання. Якщо без поважних причин студент не виконав, не оформив та не захистив завдання у вказаний термін, то подальша підсумкова оцінка за завдання буде знижена не менш ніж на 1 бал за виконання кожного завдання.

СПИСОК ЛІТЕРАТУРИ

1. Бардачев Ю. Н., Соколова Н. А., Ходаков В. Е. Основы дискретной математики. – Херсон, ХГТУ, 2000. – 356 с.
2. Виленкин Н. Я. Рассказы о множествах. – М.: МЦНМО, 2007. – 150 с.
3. Горбатов В. А. Основы дискретной математики. – М.: Высшая школа, 1986. – 312 с.
4. Касаткин В. Н. Необычные задачи математики. – К.: Радянська школа, 1987. – 125 с.: іл.
5. Капітонова Ю.В. та ін. Основи дискретної математики/ Ю. В. Капітонова, С.Л. Кривий, О.А. Летичевский та ін. – К.: Наукова думка, 2002. – 578 с.
6. Клини С. Математическая логика. – М.: Мир,1973. – 480 с.
7. Комп'ютерна дискретна математика: Підручник/ Бондаренко М. Ф., Білоус Н. В., Руткас А. Г. – Харків: Компанія СМІТ, 2004. – 480 с.
8. Кузнєцов О. П., Адельсон – Вельский Г.М. Дискретная математика для инженера. – М.: Энергия, 1980. – 344с.
9. Кузин Л. Т. Основы кибернетики: в 2-х т. – Т. 2. Основы кибернетических моделей: Учебное пособие для вузов. – М.: Энергия, 1979. – 584 с.: ил.
10. Лихтарников Л. М., Сукачева Т. Г. Математическая логика. – М.: Высшая школа, 1999. – 200 с.
11. Ловас Л., Палмер М. Прикладные задачи теории графов. Теория паросочетаний в математике , физике, химии. – М.: Мир, 1998. – 653 с.
12. Цой С., Цхай С. М. Прикладная теория графов. – Алма-Ата: Наука, 1971. – 500 с.: ил.

Методичні вказівки щодо виконання розрахунково-графічної роботи з дисципліни «Дискретна математика» зі спеціальності 123 – «Комп'ютерна інженерія»

Укладачі: старш. викл. В. Ю. Бельська,

старш. викл. А. Л. Юдіна

Відповідальний за випуск проф. А. В. Луговой

Підп. до др.________________.Формат 60×84 1/16. Папір тип. Друк ризографія.

Ум. друк. арк._____. Наклад_________прим. Зам. №___________. Безкоштовно.

Видавничий відділ КрНУ імені Михайла Остроградського

вул. Першотравнева, 20, м. Кременчук, 39614