

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

П.Г. Стеганцева, Є.В. Стеганцев

КОМП'ЮТЕРНА ДИСКРЕТНА МАТЕМАТИКА І АНАЛІЗ ДАНИХ

Методичні рекомендації
до лабораторних занять
для здобувачів ступеня вищої освіти бакалавра
спеціальності «Математика»
освітньо-професійної програми «Математика»

Затверджено
вченою радою ЗНУ
Протокол №

Запоріжжя
2025

Стеганцева П. Г., Стеганцев Є. В. Комп'ютерна дискретна математика і аналіз даних : методичні рекомендації до лабораторних занять для здобувачів ступеня вищої освіти бакалавра спеціальності «Математика» освітньо-професійної програми «Математика». Запоріжжя : Запорізький національний університет, 2025. 56 с.

У методичних рекомендаціях представлена розробка завдань до кожного лабораторного заняття курсу «Комп'ютерна дискретна математика і аналіз даних». Міститься інформація та завдання для актуалізації знань, приклади розв'язання та оформлення завдань, загальні вимоги до змісту та оформлення звіту. Запропоновані завдання сприятимуть формуванню навичок користування пакетами прикладних програм та застосування алгоритмів розв'язання задач дискретної математики на рівні програмної реалізації.

Видання призначене для здобувачів ступеня вищої освіти бакалавра спеціальності «Математика», які навчаються за освітньо-професійною програмою «Математика»..

Рецензент

Є. В. Панасенко, кандидат фізико-математичних наук, доцент кафедри фундаментальної та прикладної математики

Відповідальний за випуск

І. В. Зіновєєв, кандидат фізико-математичних наук, доцент, завідувач кафедри загальної математики

ЗМІСТ

ВСТУП.....	4
ЗАГАЛЬНІ ВИМОГИ ДО ЗМІСТУ ТА ОФОРМЛЕННЯ ЗВІТУ	6
ЗМІСТОВИЙ МОДУЛЬ 1 ОСНОВНІ АЛГОРИТМИ ТЕОРІЇ МНОЖИН, ВІДНОШЕНЬ ТА ФУНКЦІЙ.....	7
Лабораторне заняття 1 <i>Реалізація операцій над множинами</i>	7
Лабораторне заняття 2 <i>Генерування булеана скінченної множини</i>	10
Лабораторні заняття 3-4 <i>Представлення відношень у ЕОМ. Дослідження властивостей бінарних відношень</i>	12
Лабораторні заняття 5-6 <i>Алгоритми побудови замикань бінарних відношень. Алгоритм топологічного сортування</i>	16
Лабораторне заняття 7 <i>Булеві функції та їх нормальні форми</i>	19
ЗМІСТОВИЙ МОДУЛЬ 2 АЛГОРИТМИ ДОСЛІДЖЕННЯ КОМБІНАТОРНИХ ОБ'ЄКТІВ	24
Лабораторні заняття 8-9 <i>Генерування комбінаторних об'єктів</i>	24
ЗМІСТОВИЙ МОДУЛЬ 3. ОСНОВНІ ПОНЯТТЯ ТЕОРІЇ ГРАФІВ. БАЗОВІ АЛГОРИТМИ НА ГРАФАХ	28
Лабораторні заняття 10-11 <i>Представлення графів у ЕОМ. Базові алгоритми на звичайних графах</i>	28
ЗМІСТОВИЙ МОДУЛЬ 4. АЛГОРИТМИ РОЗВ'ЯЗАННЯ ОПТИМІЗАЦІЙНИХ ЗАДАЧ ТЕОРІЇ ГРАФІВ.....	34
Лабораторні заняття 12-13 <i>Алгоритми на зважених графах</i>	34
Лабораторне заняття 14 <i>Знаходження максимального потоку в мережі</i>	41
Лабораторне заняття 15 <i>Розв'язання задачі про призначення</i>	47
ВИКОРИСТАНА ЛІТЕРАТУРА	54
РЕКОМЕНДОВАНА ЛІТЕРАТУРА	55
ІНФОРМАЦІЙНІ РЕСУРСИ.....	55

ВСТУП

Дисципліна «Комп'ютерна дискретна математика і аналіз даних» є важливою частиною професійної підготовки бакалаврів за спеціальністю «Математика» освітньо-професійної програми «Математика».

Метою викладання навчальної дисципліни «Комп'ютерна дискретна математика і аналіз даних» є ознайомлення студентів з основними алгоритмами розв'язання задач сучасної дискретної математики та їх аналіз.

Основними **завданнями** вивчення дисципліни «Комп'ютерна дискретна математика» є:

- вивчення методів комбінаторного аналізу;
- ознайомлення з основними алгоритмами теорії графів;
- доведення математичного викладення математичного матеріалу до рівня реалізації програм;
- розвиток навичок формалізації та побудови математичної моделі дискретних математичних об'єктів;
- формування навичок представлення в ЕОМ основних об'єктів дискретної математики;
- розвиток навичок конструювання алгоритмів дослідження основних об'єктів дискретної математики.

Згідно з вимогами освітньої-професійної програми здобувачі освіти мають досягти таких компетентностей та програмних результатів навчання.

Компетентності:

- ЗК1. Здатність до абстрактного мислення, аналізу та синтезу;
- ЗК2. Здатність застосовувати знання у практичних ситуаціях;
- ЗК8. Здатність до пошуку, обробки та аналізу інформації з різних джерел;
- СК2. Здатність подавати математичні міркування та висновки з них у формі, придатній для цільової аудиторії, а також аналізувати та обговорювати математичні міркування інших осіб, залучених до розв'язання тієї самої задачі;
- СК5. Здатність до кількісного мислення;
- СК9. Здатність застосовувати спеціалізовані мови програмування та пакети прикладних програм;

Результати навчання:

- РН 4. Розуміти фундаментальну математику на рівні, необхідному для досягнення інших вимог освітньої програми;
- РН 5. Мати навички використання спеціалізованих програмних засобів комп'ютерної та прикладної математики і використовувати інтернет-ресурси;
- РН 10. Розв'язувати задачі придатними математичними методами, перевіряти умови виконання математичних тверджень, коректно переносити

умови та твердження на нові класи об'єктів, знаходити й аналізувати відповідності між поставленою задачею й відомими моделями;

- РН 11. Розв'язувати конкретні математичні задачі, які сформульовано у формалізованому вигляді; здійснювати базові перетворення математичних моделей;

- РН 12. Відшуковувати потрібну науково-технічну інформацію у науковій літературі, базах даних та інших джерелах інформації;

- РН 20. Розв'язувати основні математичні задачі аналізу даних; застосовувати базові загальні математичні моделі для специфічних ситуацій, мати навички управління інформацією, і застосування комп'ютерних засобів статистичного аналізу даних.

Курс «Комп'ютерна дискретна математика і аналіз даних» базується на знаннях та вміннях, отриманих під час вивчення дисциплін «Дискретна математика», «Основи програмування». Поняття й методи дискретної математики є інструментом подачі та обробки інформації. Вміння користуватись пакетами прикладних програм допоможуть застосувати алгоритми розв'язання задач дискретної математики на рівні програмної реалізації. Серед задач аналізу даних розглядатимуться: вибір способу представлення об'єктів у ЕОМ, виявлення зв'язків між об'єктами, класифікація, зниження розмірності, аналіз та інтерпретація отриманих результатів, вибір способів візуалізації даних задачі й отриманих результатів. Отримані знання та сформовані компетентності будуть важливими для успішного проходження Виробничої практики, а також для вивчення дисциплін освітнього рівня магістра, зокрема, «Машинне навчання».

Методичні рекомендації укладено відповідно до силабусу дисципліни «Комп'ютерна дискретна математика і аналіз даних».

Загальні вимоги до змісту та оформлення звіту

Звіт являє собою оформлені результати виконання завдань лабораторної роботи. Звіт до поточної лабораторної роботи надсилається у систему Moodle до наступного лабораторного заняття.

Звіт до кожної лабораторної роботи має таку структуру:

1) Титульний лист – є першою сторінкою звіту і містить інформацію:

- назва ВНЗ та факультету;
- назва дисципліни;
- тема лабораторної роботи;
- прізвище, ім'я та номер групи студента;
- прізвище викладача;
- місто та рік виконання роботи.

2) Завдання до лабораторної роботи – список всіх задач, номер варіанту.

3) Розв'язання кожної задачі – містить:

- опис і аналіз вхідних та вихідних даних – тип, обмеження;
- опис математичної моделі задачі;
- розв'язання задачі «вручну», якщо це вимагається в умові;
- результат роботи комп'ютерної програми для розв'язання задачі – скріншот результату за своїм варіантом;
- тестовий приклад для перевірки роботи програми – скріншот результату після зміни даних (на вибір: зміна номеру варіанту, зміна масиву – додавання або видалення елементів тощо).

4) Висновки, коментарі.

5) Список використаних джерел.

Змістовий модуль 1 Основні алгоритми теорії множин, відношень та функцій

Лабораторне заняття 1 *Реалізація операцій над множинами*

Мета. Ознайомитись зі способами збереження інформації про належність елемента до множини. Описати алгоритми для знаходження результатів операцій над множинами.

Необхідне обладнання та матеріали.

Лекція 1

Бондаренко М. Ф. Комп'ютерна дискретна математика : підручник для студ. вузів. Харків : Компанія СМІТ, 2004. 445 с.

Журавчак Л. М. Дискретна математика для програмістів : навч. посіб. Львів : Львівська політехніка, 2019. 420 с.

Спекторський І. Я. Дискретна математика. Збірник задач : навчальний посібник.

URL :

http://spectorsky.ho.ua/files/rasch_%D1%81%D0%B0%D0%BC%D0%A0%D0%BE%D0%B1.pdf

Актуалізація знань.

🔔 **Увага!** Обговорення питань:

- Способи задання множини;
- Означення операцій над множинами;
- Візуалізація результатів операцій над множинами;
- Двійкова система числення. Логічний добуток та логічна сума двійкових чисел.

Хід виконання роботи.



Один і той самий об'єкт може бути представлений різними способами. Вибір представлення залежить від багатьох факторів: особливостей цього об'єкта, частоти його використання в задачі тощо. Програмісту треба знати декілька способів й вміти обирати найбільш зручний для задачі, що розв'язується.

Представлення множини машинним словом або бітовою шкалою

Нехай універсальна множина скінченна й число n її елементів не перевищує розрядності ЕОМ. Занумеруємо елементи універсума символами $u_1, u_2, \dots, u_n$. Підмножина A універсума представляється кодом S , в якому

$$C[i] = \begin{cases} 1, & \text{if } u_i \in A, \\ 0, & \text{if } u_i \notin A, \end{cases}$$

де $C[i]$ – це i -тий розряд кода C .

Тоді, код перетину множин A та B є порозрядний логічний добуток кода множини A та кода множини B . Код об'єднання множин A та B є порозрядною логічною сумою кода множини A та кода множини B . Код доповнення множини A до універсума є інверсією кода множини A . У більшості ЕОМ для цих операцій є відповідні команди. В такий спосіб операції виконуються ефективно у випадку «невеликих множин»

Алгоритм злиття для знаходження перетину множин

Множини будемо зберігати як масив з нумерацією елементів, починаючи з одиниці. Позначимо через i номер поточного елемента в множині A , через j – номер поточного елемента з множини B . Будемо будувати множину P , яка є перетином множин A та B . Через k позначимо потужність множини P . Також k буде й номером останнього доданого елемента в P .

Описання алгоритму:

- 1) Покласти $i = j = 1, k = 0$.
- 2) Якщо в A та B (одночасно) є ще не переглянуті елементи, виконати:
 - Якщо $A[i] = B[j]$, то виконати:
 - i. Додати $A[i]$ в P , тобто $k := k + 1$ й $P[k] := A[i]$
 - ii. Перейти до наступних елементів множин A, B , тобто $i := i + 1$ й $j := j + 1$
 - Якщо $A[i] < B[j]$, то перейти до наступного елемента множини A , тобто $i := i + 1$
 - В протилежному випадку (тобто коли $A[i] > B[j]$) перейти до наступного елемента множини B , тобто $j := j + 1$
- 3) Перейти до пункту 2).

Приклад. $A = \{1, 3, 6, 8, 9\}, B = \{3, 4, 5, 6, 7\}$.

Розглядувані пари: (1,3), (3,3), (6,4), (6,5), (6,6), (8,7)

Відбір елементів множини P : 3 6

Відповідь: $P = \{3, 6\}$

Алгоритм знаходження об'єднання злиттям

Позначимо через i номер поточного елемента в множині A , через j – номер поточного елемента в множині B . Будемо будувати множину P , яка є об'єднанням множин A та B . Через k позначимо потужність множини P . Також k буде й номером останнього доданого в P елемента.

Описання алгоритму:

- 1) Покласти $i = j = 1, k = 0$.
- 2) Якщо є не переглянуті елементи множин A, B , виконати:

- Якщо в A ще є елементи, й в B є елементи й $A[i] = B[j]$, то
 - i. Додати $A[i]$ в P , тобто $k := k + 1$ й $P[k] := A[i]$
 - ii. Перейти до наступних елементів в A та B , тобто $i := i + 1$ й $j := j + 1$
 - Якщо в B вже всі елементи були проглянуті або ж $A[i] < B[j]$ (при умові, що в A не всі елементи були переглянуті), виконати:
 - i. Додати $A[i]$ в P , тобто $k := k + 1$ й $P[k] := A[i]$
 - ii. Перейти до наступного елемента множини A , тобто $i := i + 1$
 - У всіх інших випадках (тобто коли в A уже всі елементи переглянуто або ж якщо $A[i] > B[j]$), виконати:
 - i. Додати $B[j]$ в P , тобто $k := k + 1$ й $P[k] := B[j]$;
 - ii. Перейти до наступного елемента множини B , тобто $j := j + 1$
- 3) Перейти до пункту 2).

Як видно, на кожному кроці ми додаємо в P мінімальний елемент з $A[i]$ й $B[j]$ та переходимо до розгляду наступного елемента.

Приклад. $A = \{1, 3, 6, 8, 9\}$, $B = \{3, 4, 5, 6, 7\}$.

Порівнюємо елементи в парах: (1,3), (3,3), (6,4), (6,5), (6,6), (8,7)

Відбір елементів множини P 1 3 4 5 6 7 8 9

Відповідь: $P = \{1, 3, 4, 5, 6, 7, 8, 9\}$

Завдання лабораторної роботи

Завдання 1. Нехай U – множина цілих чисел від 0 до 25,

A – множина чисел з U , кратних n ,

B – множина чисел з U , не більших n ,

V – множина чисел з U , які є точними квадратами.

- 1) Записати коди підмножин A , B , V .
- 2) Записати коди перетину множин A та B .
- 3) Записати коди об'єднання множин B та V .
- 4) Записати код доповнення до множини A .

Завдання 2. Записати алгоритм і програму знаходження об'єднання та перетину множини $P = \{a, b, c, d, e\}$ (табл. 1) та множини K натуральних чисел від 1 до 40, застосувавши алгоритм злиття.

За допомогою алгоритма злиття з'ясувати чи є множина P підмножиною множини K .

Завдання 3. Нехай $U = \{5, 6, \dots, 30\}$ – універсальна множина,
 $A = \{5, 7, 8, 10, 11, 14, 20, 21, 22, 26\}$, $B = \{2x - 7 \mid x - \text{непарне число}\}$,
 $C = \{x \mid x^2 + 1 - \text{просте число}\}$. Перелічити елементи множини $D = B \Delta (\bar{A} \cup C)$.

Таблиця 1 – Задання елементів множини P

№	a	b	c	d	e
1	2	10	-8	-33	-18
2	-11	-30	22	95	-75
3	2	6	-4	-13	-6
4	9	-26	20	24	-32
5	3	5	-27	32	-12
6	6	-11	2	12	-8
7	7	-7	-19	16	20
8	15	-83	205	-216	80
9	5	-2	-14	3	9
10	5	-4	-16	32	-16

Лабораторне заняття 2 Генерування булеана скінченної множини

Мета. Ознайомитись з двома алгоритмами перерахування всіх підмножин заданої множини, особливостями їх застосування. Протестувати власні програми реалізації цих алгоритмів.

Необхідне обладнання та матеріали.

Лекція 1

Бондаренко М. Ф. Комп'ютерна дискретна математика : підручник для студ. вузів. Харків : Компанія СМІТ, 2004. 445 с.

Борисенко О. А. Дискретна математика : підручник для студентів вищих навчальних закладів. Суми : Університетська книга, 2019. 255 с.

Журавчак Л. М. Дискретна математика для програмістів : навч. посіб. Львів : Львівська політехніка, 2019. 420 с.

Спекторський І. Я. Дискретна математика. Збірник задач : навчальний посібник.

URL :
http://spectorsky.ho.ua/files/rasch_%D1%81%D0%B0%D0%BC%D0%A0%D0%BE%D0%B1.pdf

Актуалізація знань.

♣ **Увага!** Обговорення питань:

- Бітові шкали підмножин заданої множини;
- Відтворення підмножини за її бітовою шкалою.

Хід виконання роботи.



Алгоритм 1 генерації булеана заданої n -елементної множини

Вхід: $n \geq 0$ - потужність множини

Вихід: послідовність бінарних кодів i підмножин заданої множини

```
for  $i$  from 0 to  $2^n - 1$  do  
    yield  $i$   
end for
```

Обґрунтування алгоритму. Алгоритм видає 2^n різних цілих чисел, а отже й різних кодів. Зі збільшенням числа збільшується кількість двійкових розрядів, необхідних для його представлення. Найбільше число $2^n - 1$ потребує для свого представлення рівно n розрядів. Таким чином, всі підмножини заданої множини генеруються, причому один раз.

Зауваження. Недолік цього алгоритму полягає в тому, що порядок генерації підмножин не пов'язаний з їх складом. Наприклад, при генерації булеану множини (a, b, c, d) за підмножиною з кодом 0111 з'явиться підмножина з кодом 1000. Першій відповідає підмножина (b, c, d) , а другій – підмножина (a) . Цей недолік ускладнює розв'язання певних задач, наприклад, задач *аналізу отриманих даних* на предмет, чи задовольняють вони заданим умовам. Якщо би сусідні підмножини мало відрізнялись складом, було б можливо скористатись результатом оцінки елементів, що розглядались на попередньому кроці генерації.

Алгоритм 2 побудови бінарного коду Грея

Особливістю цього алгоритму є той факт, що при такій генерації булеана кожна наступна підмножина отримується з попередньо згенерованої видаленням або додаванням лише одного елемента.

Вхід: $n \geq 0$ – потужність множини

Вихід: послідовність бінарних кодів підмножин B заданої множини

B : array $[1 \dots n]$ of $0 \dots 1$ {бітова шкала для представлення підмножин}

```
for  $i$  from 1 to  $n$  do  
     $B[i] := 0$  {ініціалізація}  
end for  
yield  $B$  {порожня множина}  
for  $i$  from 0 to  $2^n - 1$  do  
     $p := Q(i)$  {визначення елемента, що підлягає додаванню або  
        видаленню}  
     $B[p] := 1 - B[p]$  {додавання або видалення елемента}
```

```

    yield B {чергова підмножина}
end for
proc Q(i) {ступінь 2 в розкладанні i на множники +1}
q:=1; j:=i
while j парне do
    j:=j/2; q:=q+1
end while
returne q
end proc

```

Приклад. Результат виконання алгоритмів 1 та 2 для множини $A = (a, b, c, d)$

Алгоритм 1				Алгоритм 2				
<i>i</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>i</i>	<i>p</i>	<i>a</i>	<i>b</i>	<i>c</i>
0	0	0	0	0		0	0	0
1	0	0	1	1	1	0	0	1
2	0	1	0	2	2	0	1	1
3	0	1	1	3	1	0	1	0
4	1	0	0	4	2	1	1	0
5	1	0	1	5	3	1	1	1
6	1	1	0	6	2	1	0	1
7	1	1	1	7	1	1	0	0

Завдання лабораторної роботи

Завдання 1. Нехай P – множина цілих чисел (див. таблицю). Присвоїти кожному з них в заданому порядку номер від 1 до 5. Записати алгоритм і програму генерування булеана множини P у лексикографічному порядку. Вивести на екран підмножини з порядковими номерами $n+3, n+5$.

Завдання 2. Записати алгоритм і програму побудови бінарного коду Грея для генерування булеана $B(P)$ множини P (табл. 1). Вивести на екран підмножини з порядковими номерами $n+3, n+5$.

Завдання 3. Нехай $U = \{3, 4, \dots, 15\}$ – універсальна множина, $A = \{5, 7, 9, 11, 12, 14\}$, $C = \{x \mid (x^2 - x) : 6\}$. Знайти кількість власних підмножин множини $C \Delta A$.

Лабораторні заняття 3-4 Представлення відношень у ЕОМ. Дослідження властивостей бінарних відношень

Мета. Ознайомитись зі способами представлення бінарних відношень у ЕОМ. Застосувати алгоритми для дослідження властивостей відношень.

Необхідне обладнання та матеріали.

Лекція 2

Бондаренко М. Ф. Комп'ютерна дискретна математика : підручник для студ. вузів. Харків : Компанія СМІТ, 2004. 445 с.

Журавчак Л. М. Дискретна математика для програмістів : навч. посіб. Львів : Львівська політехніка, 2019. 420 с.

Капітонова Ю. В. Основи дискретної математики : підручник. Київ : Наукова думка, 2012. 579 с.

Спекторський І. Я. Дискретна математика. Збірник задач : навчальний посібник. URL :

http://spectorsky.ho.ua/files/rasch_%D1%81%D0%B0%D0%BC%D0%A0%D0%BE%D0%B1.pdf

Актуалізація знань.

📌 **Увага!** Дискусія з питань:

- Означення й способи задання бінарного відношення;
- Приклади бінарних відношень на різних множинах;
- Властивості бінарних відношень, заданих на одній множині;
- Важливі класи бінарних відношень, заданих на одній множині.

Хід виконання роботи.



Говорять, що елементи a та b утворюють *упорядковану пару* (a, b) , якщо порядок їх запису важливий, тобто, (a, b) і (b, a) – різні упорядковані пари при умов, що a і b різні. *Декартовим добутком* двох множин A та B називається множина упорядкованих пар, в яких перший елемент належить множині A , а другий – множині B :

$$A \times B = \{(a, b) \mid a \in A, b \in B\}.$$

Нехай задано дві множини A та B . *Бінарним відношенням* R з множини A в множину B називається будь-яка підмножина декартового добутку множин A та B , отже $R \subseteq A \times B$. Якщо пара (a, b) належить бінарному відношенню R , то використовують запис aRb . Якщо $A=B$, то говорять, що R є *бінарним відношенням на множині* A .

Означення. Нехай R є бінарним відношенням на множині A . Відношення

- $R^{-1} = \{(a, b) \mid (b, a) \in R\}$ називається *оберненим*;
- $\bar{R} = \{(a, b) \mid (a, b) \notin R\}$ називається *доповненням*;
- $I = \{(a, a) \mid a \in A\}$ називається *тотожнім* або *діагоналлю*;
- $U = \{(a, b) \mid a \in A, b \in A\} = A \times A$ називається *універсальним*.

Означення. Нехай $R_1 \subseteq A \times C$ – відношення з A в C , $R_2 \subseteq C \times B$ – відношення з C в B . Композицією відношень R_1 та R_2 називається відношення $R \subseteq A \times B$ з A в B , яке визначається так:

$$R = R_1 \circ R_2 = \{(a, b) \mid a \in A \wedge b \in B \wedge \exists c \in C \ aR_1c \wedge cR_2b\}.$$

Задача 1. Знайти композицію відношень

$R_1 = \{(1,1), (1,2), (1,3), (3,1), (2,3)\}$ та $R_2 = \{(2,1), (3,3), (3,1), (2,4)\}$, заданих на множині натуральних чисел.

Означення. Відношення $R \subseteq A \times A$ називається

- Рефлексивним, якщо $\forall a \in A \ aRa$;
- Антирефлексивним, якщо $\forall a \in A \ \overline{aRa}$;
- Симетричним, якщо $\forall a, b \in A \ aRb \Rightarrow bRa$;
- Антисиметричним, якщо $\forall a, b \in A \ aRb \wedge bRa \Rightarrow a = b$;
- Транзитивним, якщо $\forall a, b, c \in A \ aRb \wedge bRc \Rightarrow aRc$;
- Повним або лінійним, якщо $\forall a, b \in A \ a \neq b \Rightarrow aRb \vee bRa$.

Теорема. Відношення $R \subseteq A \times A$

- Рефлексивне $\Leftrightarrow I \subseteq R$;
- Антирефлексивне $\Leftrightarrow R \cap I = \emptyset$;
- Симетричне $\Leftrightarrow R = R^{-1}$;
- Антисиметричне $\Leftrightarrow R \cap R^{-1} \subseteq I$;
- Транзитивне $\Leftrightarrow R \circ R \subseteq R$;
- Повним або лінійним $\Leftrightarrow R \cup I \cup R^{-1} = U$.

Задача 2. Користуючись теоремою, перевірити виконання властивостей відношення $R_1 = \{(1,1), (1,2), (1,3), (3,3), (3,1), (3,2), (2,3)\}$ на множині $A = \{1,2,3,4\}$

Бінарне відношення ρ на множині X називається *відношенням порядку*, якщо воно транзитивне й антисиметричне. Множина, на якій задано відношення порядку, називається *впорядкованою*.

Відношення порядку називається:

- відношенням часткового порядку, якщо воно неповне;
- відношенням лінійного порядку, якщо воно повне.

Відношення часткового порядку називається:

- відношенням строгого часткового порядку, якщо воно антирефлексивне,
- відношенням нестрогого часткового порядку, якщо рефлексивне

Відношення лінійного порядку називається:

- відношенням строгого лінійного порядку, якщо воно антирефлексивне,

- відношенням нестрогого лінійного порядку, якщо воно рефлексивне.

Приклади.

- відношення « $\subset$ » (бути власною підмножиною) на булеані деякої множини є відношенням строгого часткового порядку,
- відношення « $<$ » на множині R є відношенням строгого лінійного порядку,
- відношення « $\subseteq$ » (бути підмножиною) на булеані деякої множини є відношенням нестрогого часткового порядку,
- відношення « $\leq$ » на множині R є відношенням нестрогого лінійного порядку.

Представлення відношень у ЕОМ

Нехай $R \subseteq A^2$ й $|A| = n$. Перенумеруємо елементи множини A . Тоді відношення R можна представити матрицею $[R]_{ij}$ за правилом

$$R[i, j] = \begin{cases} 1, \text{ якщо } iRj, \\ 0, \text{ якщо } \overline{iRj}. \end{cases}$$

Теорема. $[R_1 \circ R_2]_{ij} = \bigvee_{k=1}^n [R_1]_{ik} \wedge [R_2]_{kj}$.

Наслідок. Матриця k -того степеня відношення R дорівнює k -тому степеню матриці цього відношення.

Задача 3. Знайти матрицю композиції відношень з задачі 1

Теорема. 1) $[R_1 \cup R_2]_{ij} = [R_1]_{ij} \vee [R_2]_{ij}$,

2) $[R_1 \cap R_2]_{ij} = [R_1]_{ij} \wedge [R_2]_{ij}$,

3) Якщо $(\forall i)(\forall j) [R_1]_{ij} \leq [R_2]_{ij}$, то $R_1 \subseteq R_2$.

Задача 4. Який вигляд має матриця відношення:

- Рефлексивного,
- Антирефлексивного,
- Симетричного,
- Антисиметричного,
- Транзитивного,
- Повного.

Завдання лабораторної роботи

Завдання 1. Побудувати матриці відношень R та B , заданих на множині $M = \{a: 9 < a < 46\}$.

А) Знайти матрицю композиції $R \circ B$,

В) Знайти, які властивості (рефлексивність, симетричність, транзитивність, антисиметричність, антирефлексивність, повнота) має відношення $R \circ B$.

Варіанти:

1. В: « x ділить y », тобто « y ділиться на x ».
2. В: « xy є парним»
3. В: « xy ділиться на 7»
4. В: « x є парним і $x+y$ є парним»
5. В: « x ділиться на 3 і $x+y$ ділиться на 3»
6. В: « $x < y$ »
7. В: « xy є непарним»
8. В: « xy ділиться на 11»
9. В: « x є парним і $x+y$ є непарним»
10. В: « x ділиться на 5 і $x+y$ ділиться на 5»

R : «різниця $x-y$ ділиться на 3» (для варіантів 1-5),

R : «різниця $x-y$ ділиться на 5» (для варіантів 6-10).

Завдання 2. Довести, що відношення R : «різниця $x-y$ ділиться на 3», задане на множині $M = \{a: 9 < a < 46\}$, є відношенням еквівалентності. Побудувати фактор-множину M / R .

Завдання 3. Для бінарного відношення $R = \{(1,2), (3,2), (9,2), (10,5), (3,3)\}$, визначеного на множині A перших 10 натуральних чисел, вказати мінімальне за кількістю елементів бінарне відношення S , яке є відношенням еквівалентності на множині A і задовольняє умову $R \subseteq S$.

Лабораторні заняття 5-6 Алгоритми побудови замикань бінарних відношень. Алгоритм топологічного сортування

Мета. Ознайомитись з алгоритмами побудови рефлексивних та транзитивних замикань бінарних відношень та з алгоритмом топологічного сортування. Створити відповідні програми та протестувати їх.

Необхідне обладнання та матеріали.

Лекція 2

Базилевич Л. Дискретна математика у прикладах і задачах : підручник.
URL : http://chyslo.com.ua/index.php?route=product/product&product_id=57

Бондаренко М. Ф. Комп'ютерна дискретна математика : підручник для студ. вузів. Харків : Компанія СМІТ, 2004. 445 с.

Журавчак Л. М. Дискретна математика для програмістів : навч. посіб. Львів : Львівська політехніка, 2019. 420 с.

Актуалізація знань.

♫ **Увага!** Дискусія з питань:

- Властивості бінарних відношень, заданих на одній множині;
- Означення замикання бінарного відношення;
- Приклади відношень та їх замикань;
- Приклади відношень часткового та лінійного порядків.

Хід виконання роботи.



Якщо відношення на множині не має ту чи іншу властивість, то це відношення можна продовжити (додати нові пари) до нового відношення, яке буде мати потрібну властивість.

Означення. Бінарне відношення R^* називається *замиканням відношення R відносно властивості P* , якщо

- R^* має властивість P ;
- $R \subset R^*$;
- R^* є підмножиною будь-якого іншого відношення, що містить R й має властивість P .

Наприклад, якщо у якості властивості P розглянути транзитивність, то матимемо транзитивне замикання.

Алгоритм Уоршалла (знаходження транзитивного замикання)

Задача знаходження транзитивного замикання відношення рівносильна задачі знаходження матриці досяжності орієнтованого графу.

В ході роботи алгоритму будується послідовність матриць $W_0, W_1, \dots, W_n$, остання є матрицею транзитивного замикання відношення.

При обчисленні матриці W_k треба:

- 1) Взяти k -тий стовпець матриці W_{k-1} ;
- 2) Рядок з номером $i, i = \overline{1, n}$, в якому на k -тому місці стоїть 0, переписати в i -тий рядок матриці W_k ;
- 3) Знайти диз'юнкцію рядка з номером $i, i = \overline{1, n}$, в якому на k -тому місці стоїть 1, з k -тим рядком і результат записати i -тим рядком матриці W_k .

Алгоритм топологічного сортування

Доведено, що будь-який частковий порядок може бути продовжений до лінійного порядку, тобто може бути включений до лінійного порядку (в сенсі включення бінарних відношень). Доведенням є існування алгоритму його побудови. Він називається *алгоритмом топологічного сортування*.

Приклад застосування алгоритму топологічного сортування до такої задачі:

Для отримання диплома магістра біології студенту необхідно засвоїти вісім курсів з ієрархією їх вивчення, поданою в таблиці.

	Назва курсу	Попередні курси
A	Біотехнологія	B
B	Початковий курс біотехнології	C
C	Цитологія	H
D	Структура ДНК	C
E	Ензимологія	D,G
F	Дієтологія	E
G	Генна інженерія	C
H	Біологія людини	<i>Без вимог</i>

Нехай студент має намір визначити порядок, в якому він буде вивчати предмети, враховуючи їх залежність один від одного. Це можна зробити за допомогою *алгоритму топологічного сортування*.

Алгоритм створює послідовність узгоджених міток для вершин безконтурного орграфу таким чином, що коли $1, 2, 3, \dots, n$ – мітки вершин і (u, v) – дуга орграфу з вершини u з міткою i до вершини v з міткою j , то $i < j$.

Розв'язання.

Крок 0 Відношення складається з пар

$(B, A), (C, B), (H, C), (C, D), (D, E), (G, E), (E, F), (C, G), (0, H)$

Крок 1 Призначаємо мітку 1 елементу H (бо є пара $(0, H)$), замінюємо H на 0 у всіх інших парах. Маємо

$(B, A), (C, B), (0, C), (C, D), (D, E), (G, E), (E, F), (C, G)$

Крок 2 Призначаємо мітку 2 елементу C та в решті пар робимо заміну C на 0:

$(B, A), (0, B), (0, D), (D, E), (G, E), (E, F), (0, G)$

Крок 3 Наступна мітка 3 може бути призначена або елементу B, або G, або D. Нехай ми вибрали B.

$(0, A), (0, D), (D, E), (G, E), (E, F), (0, G)$

Крок 4 Мітку 4 призначимо елементу A: $(0, D), (D, E), (G, E), (E, F), (0, G)$

Крок 5 Мітку 5 призначимо елементу D: $(0, E), (G, E), (E, F), (0, G)$

Крок 6 Мітку 6 призначимо елементу G: $(0, E), (0, E), (E, F)$

Крок 7 Мітку 7 призначимо елементу E: $(0, F)$

Крок 8 Мітку 8 призначимо елементу F.

Отже, один із можливих списків курсів має вигляд:

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

Реалізація алгоритму у випадку матричного представлення відношення

- 1) Знайти в матриці відношення перший рядок з усіма нулями.
- 2) Присвоїти елементу, який відповідає цьому рядку, №1, а всі 1 у відповідному стовпці замінити на 0. Більше цей рядок не розглядати.
- 3) На наступному кроці знову шукати рядок лише з нулями. Алгоритм закінчує роботу, коли всі елементи множини отримують номери. Шуканий лінійний порядок являє собою послідовність елементів множини, виписану в порядку спадання їх номерів.

Зауваження. Якщо вихідне відношення було нестрогим, то при складанні його матриці діагональні елементи ігноруються.

Завдання лабораторної роботи

Завдання 1. Побудувати рефлексивне замикання бінарного відношення $R = \{(a,b), (b,c), (c,a)\}$, визначеного на множині $A = \{a,b,c,d\}$.

Завдання 2. Довести, що відношення $R = \{(1,1), (n+1,2), (3,1), (2,n+3), (n+1,n+3)\}$, задане на множині натуральних чисел від 1 до $n+3$, де n – номер варіанту, є відношенням часткового порядку. За допомогою алгоритму топологічного сортування доповнити його до відношення лінійного порядку на цій множині.

Завдання 3. За допомогою алгоритму Уоршала побудувати транзитивне замикання відношення $R = \{(1,1), (n+1,2), (1,3), (3,1), (2,n+3)\}$, заданого на множині натуральних чисел від 1 до $n+3$, де n – номер варіанту.

Лабораторне заняття 7 Булеві функції та їх нормальні форми

Мета. Ознайомитись зі способами представлення функцій, зокрема булевих, у ЕОМ. Застосувати алгоритми і програми для побудови досконалих нормальних форм та полінома Жегалкіна булевих функцій.

Необхідне обладнання та матеріали.

Лекція 2

Бардачов Ю. М. Дискретна математика : підруч. для студ. вищ. техн. навч. закл. Київ : Вища школа, 2007. 548 с

Бондаренко М. Ф. Комп'ютерна дискретна математика : підручник для студ. вузів. Харків : Компанія СМІТ, 2004. 445 с.

Борисенко О. А. Дискретна математика : підручник для студентів вищих навчальних закладів. Суми : Університетська книга, 2019. 255 с.

Журавчак Л. М. Дискретна математика для програмістів : навч. посіб. Львів : Львівська політехніка, 2019. 420 с.

Стеганцева П. Г., Гречнева М. О., Манько Н. І., Спиця О. Г., Стеганцев Є. В. Дискретна математика: навчальний посібник для здобувачів ступеня бакалавра освітньо-професійних програм «Середня освіта (математика)», «Середня освіта (інформатика)», «Математика», «Комп'ютерна математика», «Комп'ютерне моделювання», «ІСТ». Запоріжжя: ЗНУ, 2021. 178 с.

Актуалізація знань.

♣ **Увага!** Дискусія з питань:

- Одновимірні та багатовимірні масиви, приклади;
- Область визначення, область значень, область істинності булевої функції;
- Нормальні форми булевої функції;
- Поліном Жегалкіна булевої функції та методи його побудови.

♣ **Увага!** Інформація про відому з теорії чисел гіпотезу (*для завдання 3*):

Гіпотеза $3n+1$. Для будь-якого натурального числа n , якщо послідовно виконати такі дії:

- При парному n ділим число n на 2,
 - При непарному – знаходимо число $3n+1$,
- то завжди через скінченне число кроків в кінці послідовності отримаємо 4, 2, 1.

Хід виконання роботи.



Під *відображенням* з множини A у множину B розуміють будь-яке бінарне відношення f з A в B , яке задовольняє умову:

$$(\forall x \in A)(\forall y_1, y_2 \in B) ((x, y_1) \in f \wedge (x, y_2) \in f \Rightarrow y_1 = y_2).$$

Якщо крім цього виконується умова $(\forall x \in A)(\exists y \in B)(x, y) \in f$, то f називається *всюду визначеним відображенням* або *функцією*.

Якщо f – функція, то замість $(x, y) \in f$ пишуть $f(x) = y$ або $x \mapsto y$.

Досконалі нормальні форми булевої функції

Досконалою диз'юнктивною нормальною формою (ДДНФ) булевої функції від n змінних називається диз'юнкція всіх тих елементарних кон'юнкцій довжини n (*конституент 1*), які відповідають булевим векторам з області істинності булевої функції.

Наприклад, формули

1) $f(x_1, x_2) = (x_1 \wedge x_2) \vee (\bar{x}_1 \wedge x_2) \vee (\bar{x}_1 \wedge \bar{x}_2);$

2) $f(x_1, x_2, x_3) = (x_1 \wedge x_2 \wedge \bar{x}_3) \vee (\bar{x}_1 \wedge x_2 \wedge x_3);$

3) $f(x) = x.$

є ДДНФ.

Теорема. Для кожної булевої функції, відмінної від константи 0, існує єдина ДДНФ.

Досконалою кон'юнктивною нормальною формою (ДКНФ) булевої функції від n змінних (**конституент**) називається кон'юнкція всіх тих елементарних диз'юнкцій довжини n , які відповідають булевим векторам, що не входять в область істинності булевої функції.

Наприклад, формули

$$1) \quad f(x_1, x_2) = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_2);$$

$$2) \quad f(x_1, x_2, x_3) = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee x_3);$$

$$3) \quad f(x) = x.$$

є ДКНФ.

Теорема. Для кожної булевої функції від n змінних, відмінної від константи 1, існує єдина ДКНФ.

Алгоритм побудови ДДНФ (ДКНФ) за таблицею істинності

- 1) Виділити булеві вектори, на яких функція приймає значення 1 (значення 0);
- 2) Записати відповідні цим векторам конституенти 1 (конституенти 0);
- 3) Записати диз'юнкцію (кон'юнкцію) отриманих конституент 1 (конституент 0).

Знаходження ДДНФ та ДКНФ за допомогою логічних законів

Нехай булева функція задана формулою. Як побудувати її досконалі нормальні форми, не переходячи до таблиць істинності. Розв'язання цієї задачі можна розбити на такі кроки.

- 1) Формулу, яка задає булеву функцію, зведемо до будь-якої ДНФ (КНФ);
- 2) Якщо в цій ДНФ (КНФ) існує елементарна кон'юнкція (диз'юнкція), яка не містить хоча б однієї із змінних, наприклад змінної x_i , то помножимо (логічно помножимо) цю кон'юнкцію на $x_i \vee \bar{x}_i = 1$ (додамо до цієї диз'юнкції $x_i \bar{x}_i = 0$) та застосуємо закони дистрибутивності.

Скінчене число кроків забезпечить нам перехід до ДДНФ (ДКНФ) заданої булевої функції.

Будемо розглядати лише такі елементарні кон'юнкції (іноді їх називають *монотонними*), в яких відсутні заперечення змінних. Константу 1 теж будемо вважати такою кон'юнкцією.

Поліномом Жегалкіна булевої функції називається формула виду

$K_1 \oplus K_2 \oplus \dots \oplus K_m$, де $m \geq 1$ і $K_i, i = 1, m$ - монотонна елементарна кон'юнкція.

Теорема. Для будь-якої булевої функції існує, причому єдиний, поліном Жегалкіна.

Існує кілька *методів* знаходження поліному Жегалкіна:

- 1) Безпосереднє перетворення формули за допомогою логічних законів: $x \oplus (y \oplus z) = (x \oplus y) \oplus z$; $x \oplus y = y \oplus x$; $x \oplus 0 = x$; $x \oplus x = 0$; $x(y \oplus z) = xy \oplus xz$; $\bar{x} = x \oplus 1$; $x \vee y = xy \oplus x \oplus y$;
- 2) Заміною в ДДНФ функції символу $\vee$ символом $\oplus$ і застосування логічних законів: $\bar{x} = x \oplus 1$ та $x(y \oplus z) = xy \oplus xz$;
- 3) Методом невизначених коефіцієнтів.

Представлення функцій у ЕОМ

Нехай $f: A \rightarrow B$ є функцією одного аргументу, причому A скінченна множина (тобто $|A| = n$) і «не дуже велика».

Найбільш загальним представленням такої функції є масив **array [A] of B**, де A – тип даних, які представляють елементи множини A , а B – тип даних, які представляють елементи множини B . Якщо серед програмування допускає лише масиви з натуральними індексами, тоді елементи множини A нумеруються: $A = \{a_1, a_2, \dots, a_n\}$ і функція представляється у вигляді масиву **array [1,2,...,n] of B**.

Функція кількох аргументів представляється багатовимірним масивом. Представлення функції за допомогою масиву є ефективним (за кількістю часу), оскільки реалізація масиву частіше всього вимагає сталого часу, не залежить від розміру масиву.

У випадку «великої» або нескінченної множини A цей спосіб стає неефективним з точки зору економії пам'яті. В цьому випадку в деяких мовах програмування означення функції вводяться ключовим словом **function**.

Завдання лабораторної роботи

Завдання 1. Булеві функції задано номерами конститuent одиниці:

Варіант	
1	$f(x_1, x_2, x_3, x_4) = (0,1,2,4,5,7,8,10,11,13,14,15)$
2	$f(x_1, x_2, x_3, x_4) = (0,1,2,3,4,5,7,8,10,11,13,14)$
3	$f(x_1, x_2, x_3, x_4) = (0,1,2,3,5,6,7,9,10,11,13,14,15)$
4	$f(x_1, x_2, x_3, x_4) = (1,2,3,4,6,7,8,9,11,12,14,15)$
5	$f(x_1, x_2, x_3, x_4) = (1,3,4,5,6,8,9,10,12,13,15)$
6	$f(x_1, x_2, x_3, x_4) = (2,4,5,6,7,8,9,11,12,13,14)$
7	$f(x_1, x_2, x_3, x_4) = (0,1,2,4,5,6,10,11,12,13,14,15)$
8	$f(x_1, x_2, x_3, x_4) = (0,1,2,3,4,5,7,11,12,13,14,15)$

9	$f(x_1, x_2, x_3, x_4) = (0, 2, 3, 4, 5, 7, 8, 9, 11, 13, 14, 15)$
10	$f(x_1, x_2, x_3, x_4) = (0, 2, 3, 4, 5, 7, 9, 10, 11, 13, 14, 15)$

Мінімізувати їх методом діаграм Карно, Вейча.

Завдання 2. Подати булеву функцію у вигляді

А) ДДНФ (номери наборів, на яких функція дорівнює одиниці, вказані в колонці 1 таблиці),

Б) ДКНФ (вектор значень функції вказано в колонці 2, замість «зірок» поставити 0),

В) поліному Жегалкіна (формула функції вказана в колонці 3).

Варіант	1	2	3
1	3,6,7,9,10,11,15	111*00**01111010	$f = (A \leftrightarrow B) \vee C \rightarrow A$
2	2,3,5,7,11,14,15	1011*1000110**01	$f = A \leftrightarrow B \vee C \rightarrow B$
3	0,1,4,10,11,12,15	11000111*00011*0	$f = A \leftrightarrow B \rightarrow C \vee A$
4	2,3,6,8,9,13,14	*101*011*0001001	$f = (A \leftrightarrow B) (C \vee A)$
5	2,3,5,7,10,11,15	00100011**010011	$f = A \leftrightarrow (B C) \rightarrow B$
6	0,3,5,7,10,14,15	11*1000111*00011	$f = (A \rightarrow B) \wedge C \leftrightarrow A$
7	0,2,5,8,13,14,15	*0111**001*0011*	$f = ((A \leftrightarrow B) C) \vee A$
8	0,3,5,7,8,11,12	0*01101100011**0	$f = (A \rightarrow B) \rightarrow C \vee A$
9	1,2,3,7,11,12,15	1*000110111000*	$f = (A \rightarrow B) \leftrightarrow (C A)$
10	0,1,2,3,6,8,15	10011010111**000	$f = A \leftrightarrow B \wedge (C \rightarrow A)$

Завдання 3. Отримати послідовності для чисел від $n+5$ до $n+10$, знайти серед чисел від $n+5$ до $n+10$ число s найбільшою довжиною послідовності й число z з максимальним членом послідовності, про яку йдеться у відомій з теорії чисел гіпотезі $3n+1$.

Завдання 4. Для бінарного відношення $R = \{(1, b), (5, d), (4, b)\} \subset \{1, 2, 3, 4, 5\} \times \{a, b, c, d\}$ побудувати мінімальне за кількістю елементів функціональне відношення f , яке задовольняє умову $R \subseteq f$ та є:

А) ін'єктивним;

Б) сюр'єктивним;

В) бієктивним.

Змістовий модуль 2 Алгоритми дослідження комбінаторних об'єктів

Лабораторні заняття 8-9 Генерування комбінаторних об'єктів

Мета. Використати способи представлення множин у ЕОМ для розв'язання перерахункових комбінаторних задач. Застосувати алгоритми генерування всіх перестановок елементів даної скінченної множини й протестувати їх на множинах з невеликою кількістю елементів.

Необхідне обладнання та матеріали.

Лекції 3-4

Бондаренко М. Ф. Комп'ютерна дискретна математика : підручник для студ. вузів. Харків : Компанія СМІТ, 2004. 445 с.

Журавчак Л. М. Дискретна математика для програмістів : навч. посіб. Львів : Львівська політехніка, 2019. 420 с.

Капітонова Ю. В. Основи дискретної математики : підручник. Київ : Наукова думка, 2012. 579 с.

Скрябіна А. В., Стеганцева П. Г. Перерахування усіх То-топологій, заданих одним вектором, та підрахунок То-топологій певної ваги. Матеріали XXIII Міжнародного науково-практичного семінару ім. А.Я.Петренюка, присвяченому 70-річчю Льотної академії Національного авіаційного університету «Комбінаторні конфігурації та їхні застосування». Запоріжжя-Кропивницький : ПП Ексклюзив-Систем, 2021. С. 165-168.

Актуалізація знань.

♣ **Увага!** Дискусія з питань:

- Поняття перестановки. Транспозиція елементів перестановки;
- Відношення «менше», «більше», «дорівнює» на множині упорядкованих n -ок (перестановок);
- Лексикографічний (словниковий) порядок перерахування підмножин заданої множини;

Хід виконання роботи.



Розглянемо **три варіанти** перерахування перестановок. Нехай множина запам'ятовується у вигляді масиву $P[1], P[2], \dots, P[n]$. Всі варіанти базуються на операції поелементної транспозиції, тобто обміну значеннями змінних

1) **Лексикографічний порядок** перерахування. Фактично працюємо з індексами елементів, а не з самими елементами. Правило порівняння перестановок:

$$(x_1, x_2, \dots, x_n) < (y_1, y_2, \dots, y_n) \Leftrightarrow \exists k \geq 1 \quad (x_k < y_k \wedge x_i = y_i \quad \forall i < k).$$

Зауваження. Якщо числа замінити буквами алфавіту, то лексикографічний порядок визначить послідовність, в якій слова довжини n розташовані у словнику.

$$(21435) < (21534)$$

2) *Антилексикографічний порядок.*

Правило порівняння перестановок:

$$(x_1, x_2, \dots, x_n) < (y_1, y_2, \dots, y_n) \Leftrightarrow \exists k \geq 1 \quad (x_k > y_k \wedge x_i = y_i \quad \forall i > k).$$

Приклад. Лексикографічний порядок (перший стовпчик) та антилексикографічний порядок (другий стовпчик) перерахування перестановок довжини три:

123	123
132	213
213	132
231	312
312	231
321	321

Властивості антилексикографічного порядку:

А) В першій перестановці елементи йдуть в зростаючій послідовності, в останній – в спадній.

В) Послідовність можна розділити на n блоків довжини $(n-1)!$, які відповідають спадаючим значенням елемента в останній позиції. Перші $n-1$ позицій блока з елементом p на останній позиції визначають послідовність перестановок множини $X \setminus \{p\}$ в антилекс. порядку.

3) *З однократною транспозицією сусідніх елементів*

Проілюструємо на прикладі. Будемо вважати, що послідовність перестановок елементів $2, 3, \dots, n$ уже побудована. Тоді шукаємо послідовність отримаємо вставляючи елемент 1 всіма можливими способами в кожен з перестановок елементів множини $2, 3, \dots, n$. При $n=3$ отримаємо

23
32
123
213
231
321
312
132

Генерування k -елементних підмножин n -елементної множини

Зафіксуємо спочатку, що кількість k -елементних підмножин n -елементної множини дорівнює C_n^k .

Наведемо алгоритм генерування всіх таких підмножин у лексикографічному порядку:

```
begin
  for i:=1 to k do A[i]:=i; (перша підмножина)
  p:=k;
  while p>=1 do
    begin write (A[1],...,A[k]);
      if A[k]=n then p:=p-1 else p:=k;
      if p>=1 then
        for i:=k down to p do A[i]:=A[p]+i-p+1
    end
  end
```

Розгляньте приклад генерування всіх 4-елементних підмножин 6-елементної множини (1,2,3,4,5,6) у лексикографічному порядку, який містить помилки:

1	2	3	4
1	2	3	5
1	2	3	6
1	2	4	5
1	2	4	6
1	2	5	6
1	2	4	5
1	3	4	6
1	3	5	6
1	4	5	6
2	3	4	5
2	5	4	6
2	3	5	6
2	4	5	6
3	4	5	6

Знайдіть та виправте помилки, користуючись описанням алгоритму.

Завдання лабораторної роботи

Завдання 1. Реалізувати алгоритм перерахування всіх перестановок:

- 1) в лексикографічному порядку
 - 2) в антилексикографічному порядку
 - 3) за мінімальне число транспозицій сусідніх елементів.
- Протестувати програму для перестановок елементів з таблиці

№	a	b	c	d	e
1	2	10	-8	-33	-18
2	-11	-30	22	95	-75
3	2	6	-4	-13	-6
4	9	-26	20	24	-32
5	3	5	-27	32	-12
6	6	-11	2	12	-8
7	7	-7	-19	16	20
8	15	-83	205	-216	80
9	5	-2	-14	3	9
10	5	-4	-16	32	-16

Завдання 2. Реалізувати алгоритм генерування всіх k -елементних підмножин множини $\{1, 2, \dots, n\}$ в лексикографічному порядку. Протестувати програму для триелементних підмножин п'ятиелементної множини (див. таблицю).

Змістовий модуль 3. Основні поняття теорії графів. Базові алгоритми на графах

Лабораторні заняття 10-11 Представлення графів у ЕОМ. Базові алгоритми на звичайних графах

Мета. Ознайомитись з чотирма способами задання графів у ЕОМ, їх перевагах та недоліках. Застосувати алгоритми пошуку «у глибину» та «у ширину» для побудови кістякових дерев звичайного графа.

Необхідне обладнання та матеріали.

Лекція 5

Бондаренко М. Ф. Комп'ютерна дискретна математика : підручник для студ. вузів. Харків : Компанія СМІТ, 2004. 445 с.

Борисенко О. А. Дискретна математика : підручник для студентів вищих навчальних закладів. Суми : Університетська книга, 2019. 255 с.

Журавчак Л. М. Дискретна математика для програмістів : навч. посіб. Львів : Львівська політехніка, 2019. 420 с.

Стеганцева П. Г., Артеменко А. О. Рекурентні співвідношення для числа неізоморфних (n,m) -графів. *Вісник Запорізького національного університету. Фізико-математичні науки*. 2021. № 1. С. 51-56. URL: <https://doi.org/10.26661/2413-6549-2021-1-06>.

Актуалізація знань.

♣ **Увага!** Дискусія з питань:

- Матриці, асоційовані з графом;
- Відмінності нумерації вершин графа (алгоритми пошуку «в глибину» та «в ширину»);
- Орієнтовані й неорієнтовані графи;
- Дерева й кістякові дерева графа.

Хід виконання роботи.



Машинне представлення орієнтованих та неорієнтованих графів. Базові алгоритми

Позначення:

$G = (V, E)$, $E \subseteq V \times V$ для орієнтованого графа,

$G = (V, E)$, $E \subseteq \{\{x, y\} \mid x, y \in V \wedge x \neq y\}$ для неорієнтованого графа.

$|V| = n, |E| = m$.

Зображення графа на площині у вигляді точок (вершин графа) та з'єднуючих їх ліній (ребер графа), яке ми використовуємо при роз'язанні

«вручну» задач на графи, повністю не придатне при розв'язанні задач на графи за допомогою ЕОМ.

Вибір способу представлення графа у ЕОМ тісно пов'язаний з ефективністю алгоритмів, кожен спосіб має переваги й недоліки.

1-й спосіб – матрицею інцидентності – прямокутна матриця с n рядками (відповідає кількості вершин) та m стовпцями (відповідає кількості ребер).

Для орієнтованого графа (рис.1) в (x,y) -стовпці в рядку x стоїть -1, в рядку y стоїть 1, всі решта елементів дорівнюють 0. Для неорієнтованого графа в (x,y) -стовпці в рядках x та y стоять 1, всі решта елементів дорівнюють 0 (рис.2):

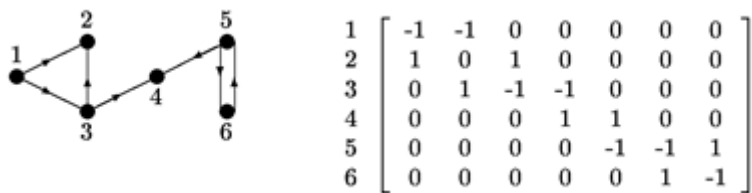


Рисунок 1 – Орієнтований граф та його матриця інцидентцій

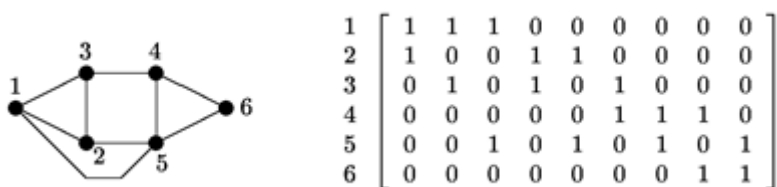


Рисунок 2 – Неорієнтований граф та його матриця інцидентцій

З алгоритмічної точки зору матриця інцидентцій є, мабуть, найгіршим способом представлення графа. По-перше, він вимагає mn комірок пам'яті, причому більшість з них зайнята нулями. Не дуже зручним є також доступ до інформації. Щоб відповісти на просте питання «чи існує дуга (x,y) ?» треба у гіршому випадку перебрати всі стовпці матриці.

Але все ж для деяких задач цей спосіб використовується.

2-й спосіб

Більш зручним є представлення графа **матрицею суміжності** – квадратною $(n \times n)$ -матрицею $B = [b_{ij}]$, де $b_{ij} = 1$, якщо існує ребро (i, j) і $b_{ij} = 0$ у противному випадку (для орієнтованого графа). У випадку неорієнтованого графа ця матриця симетрична, тому що наявність ребра $\{i, j\}$ передбачає також наявність ребра $\{j, i\}$ (рис.3).

.	1	2	3	4	5	6	.	1	2	3	4	5	6
1	0	1	1	0	0	0	1	0	1	1	0	1	0
2	0	0	0	0	0	0	2	1	0	1	0	1	0
3	0	1	0	1	0	0	3	1	1	0	1	0	0
4	0	0	0	0	0	0	4	0	0	1	0	1	1
5	0	0	0	1	0	1	5	1	1	0	1	0	1
6	0	0	0	0	1	0	6	0	0	0	1	1	0

Рисунок 3 – Матриці суміжності для графів на рис.1 та рис.2

Основною перевагою матриці суміжності є той факт, що лише за один крок можна отримати відповідь на питання «чи існує дуга (ху)? (або ребро $\{x, y\}$)». **Недолік** полягає в тому, що незалежно від числа ребер об'єм зайнятої пам'яті складає n^2 . Один із шляхів подолання для «невеликих n » – зберігати цілий рядок (або стовпчик) матриці в одному машинному слові.

3 спосіб

Більш економно (особливо у випадку «нешільних графів» – коли m значно менше за n^2) є **метод представлення графа** за допомогою **списка пар**, які відповідають ребрам графа (рис.4). Об'єм пам'яті складає $2m$.

1	1	3	3	5	5	4	1	1	1	2	2	3	4	4	5
2	3	2	4	4	6	5	2	3	5	3	5	4	5	6	6

Рисунок 4 – Списки ребер графів на рис.1 та рис.2

Недоліком є велика кількість кроків – порядку m у гіршому випадку – необхідних для отримання множини вершин, до яких ідуть ребра з данної вершини. Ситуація значно покращиться, якщо впорядкувати множину пар лексикографічно й застосувати двійковий пошук.

4 спосіб

Найкращим у багатьох випадках є структура даних, яку називають **списками інцидентій**.

Ця структура для кожної вершини $v \in V$ містить список таких вершин u , що $v \rightarrow u$ (або $v - u$ для неорієнтованого графа) (рис.5)

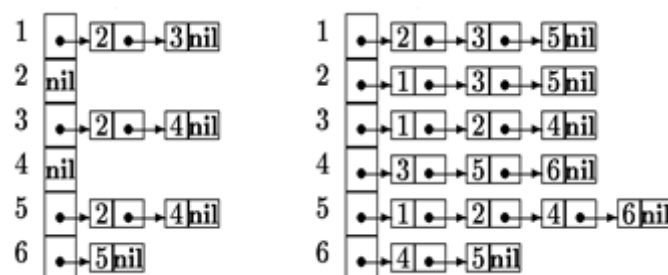


Рисунок 5 – Списки інцидентій графів на рис.1 та рис.2

Число комірок пам'яті, необхідне для представлення графа списками інцидентності дорівнює $m+n$.

При пошуку в глибину в якості активної вибирається та з відкритих вершин, яка була відвідана останньою. Найбільш зручною структурою для збереження множини відкритих вершин є **стек**: вершини, що відкриваються, складаються в стек в тому порядку, в якому вони відвідуються, а в якості активної вибирається остання вершина.

При пошуку в ширину в якості активної вибирається та з відкритих вершин, яка була відвідана раніше інших. Зручно використовувати чергу для зберігання відкритих вершин. Після відвідування вершини, вона помічається (стає не новою) й поміщається в чергу (рис. 6).

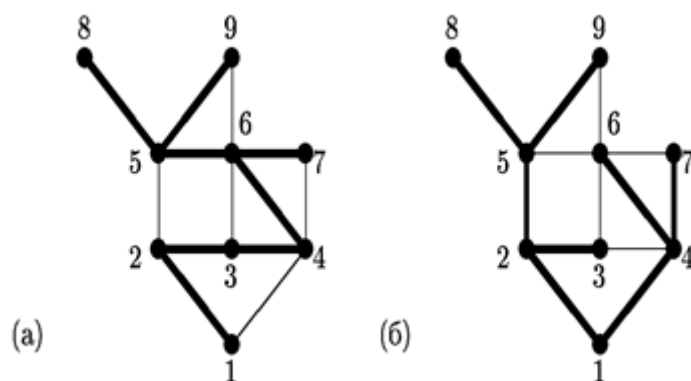


Рисунок 6 – Кістякові дерева, побудовані пошуком «в глибину» (а) та «в ширину» (в)

Порядок перегляду вершин при побудові кістякового дерева пошуком «в глибину»: **1,2,3,4,6,7,5,8,9.**

Приклад запису черги (в дужках). **Порядок** (жирним) перегляду вершин при побудові кістякового дерева пошуком «в ширину»:

1(2,4),1,2(3,5),2,4(6,7),4,3,5(8,9),5,6,7,8,9

Завдання лабораторної роботи

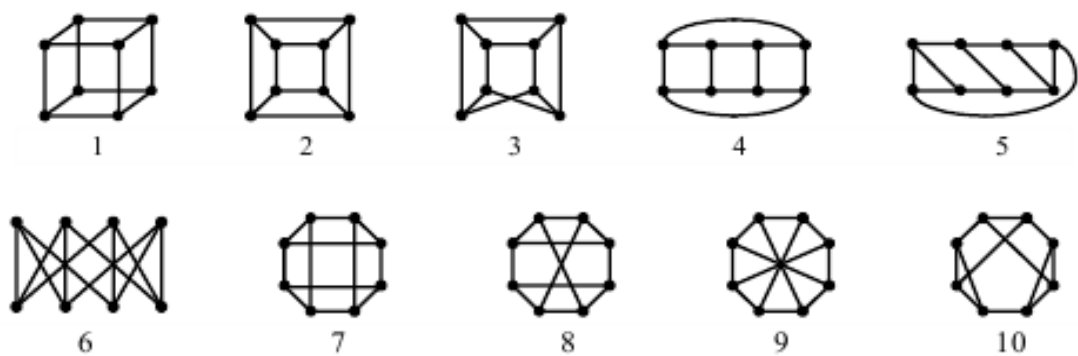
Завдання 1. Граф задано матрицею інцидентцій. Створити програму перевірки належності дуги до графа (10 варіантів).

Варіант1							
x_1	1	0	0	1	0	0	-1
x_2	-1	-1	0	0	1	0	0
x_3	0	1	-1	0	0	-1	0
x_4	0	0	0	-1	0	1	0
x_5	0	0	1	0	-1	0	1
Варіант2							

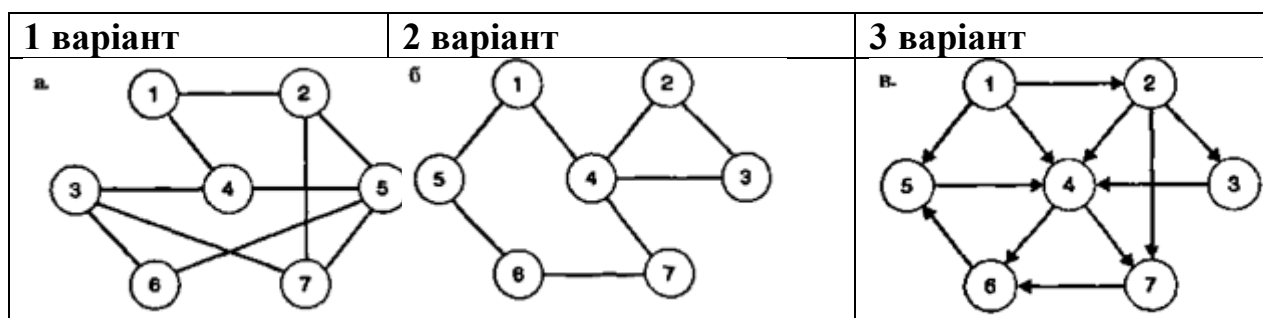
x_1	1	0	0	-1	0	0	1
x_2	0	1	0	0	1	0	0
x_3	-1	-1	1	0	0	1	0
x_4	0	0	0	1	0	-1	0
x_5	0	0	-1	0	-1	0	-1
Вариант3							
x_1	1	1	0	0	-1	0	0
x_2	0	0	1	0	0	1	0
x_3	0	0	-1	1	0	0	1
x_4	-1	0	0	0	1	0	-1
x_5	0	-1	0	-1	0	-1	0
Вариант4							
x_1	1	0	1	0	0	-1	0
x_2	0	0	0	1	0	0	1
x_3	0	1	0	-1	1	0	0
x_4	0	-1	0	0	0	1	0
x_5	-1	0	-1	0	-1	0	-1
Вариант5							
x_1	0	0	0	1	0	0	-1
x_2	1	1	0	0	1	0	0
x_3	-1	0	1	0	-1	1	0
x_4	0	0	-1	0	0	0	1
x_5	0	-1	0	-1	0	-1	0
Вариант6							
x_1	0	-1	0	0	1	0	0
x_2	-1	0	1	0	0	1	0
x_3	0	0	0	1	0	-1	1
x_4	1	1	0	-1	0	0	0
x_5	0	0	-1	0	-1	0	-1
Вариант7							
x_1	0	0	-1	0	0	1	0
x_2	-1	0	0	1	0	0	1
x_3	0	1	0	0	1	0	-1
x_4	0	0	1	0	-1	0	0
x_5	1	-1	0	-1	0	-1	0
Вариант8							
x_1	0	0	0	-1	0	0	1
x_2	0	1	0	0	1	0	0

x_3	-1	-1	1	0	0	1	0
x_4	1	0	0	1	0	-1	0
x_5	0	0	-1	0	-1	0	-1
Варіант9							
x_1	0	1	0	0	-1	0	0
x_2	0	0	1	0	0	1	0
x_3	-1	0	-1	1	0	0	1
x_4	0	0	0	0	1	0	-1
x_5	1	-1	0	-1	0	-1	0
Варіант10							
x_1	0	-1	0	0	1	0	0
x_2	0	0	1	0	0	1	0
x_3	0	0	0	1	0	-1	1
x_4	-1	1	0	-1	0	0	0
x_5	1	0	-1	0	-1	0	-1

Завдання 2. Указати порядок відвідування вершин графа при обходах «в глибину» й «в ширину». Задати отриманий помічений граф списками інциденцій (10 варіантів).



Завдання 3. Побудувати остовне дерево заданого графа методом пошуку «в глибину» й «в ширину».



4 варіант	5 варіант	6 варіант	
1. 			
7 варіант	8 варіант	9 варіант	10 варіант

Завдання 4. Для графа з *Завдання 1* за допомогою алгоритма Уоршала вирішити питання про досяжність вершини x_5 з вершини x_3 .

Змістовий модуль 4. Алгоритми розв'язання оптимізаційних задач теорії графів

Лабораторні заняття 12-13 *Алгоритми на зважених графах*

Мета. Ознайомитись з поняттями зваженого графа (неорієнтованого й орієнтованого) та способами його представлення у ЕОМ. Застосувати алгоритми побудови мінімального кістякового дерева у зваженому графі. Засвоїти алгоритм Дейкстри пошуку мінімального шляху між парою вершин у зваженому орієнтованому графі.

Необхідне обладнання та матеріали.

Лекція 6

Бондаренко М. Ф. Комп'ютерна дискретна математика : підручник для студ. вузів. Харків : Компанія СМІТ, 2004. 445 с.

Журавчак Л. М. Дискретна математика для програмістів : навч. посіб. Львів : Львівська політехніка, 2019. 420 с.

Спекторський І. Я. Дискретна математика. Збірник задач : навчальний посібник.

URL : http://spectorsky.ho.ua/files/rasch_%D1%81%D0%B0%D0%BC%D0%A0%D0%BE%D0%B1.pdf

Актуалізація знань.

♣ **Увага!** Дискусія з питань:

- Зважені графи як моделі задач практичного змісту;
- Орієнтовані графи як моделі задач практичного змісту;
- Прикладне значення задачі знаходження кістякових дерев мінімальної ваги у зваженому графі.

Хід виконання роботи.



Поставимо у відповідність кожному ребру e графа G дійсне число (вагу ребра) $\omega(e)$. Позначимо цей граф (G, ω) і назовемо **зваженим графом**.

В прикладній задачі, математичною моделлю якої є граф, вагою ребра може бути його довжина, вартість перевезення, вартість палива тощо. Як правило, цікавим є розв'язок задачі, який забезпечує мінімальні витрати ресурсів. В термінах графів ця вимога еквівалентна знаходженню в графі остовного дерева з мінімальною вагою. Очевидно, можна побудувати всі дерева (їх кількість легко знаходиться за допомогою матриці Кіркгофа даного графа), порахувати вагу кожного з них і вибрати дерево з мінімальною вагою, але це алгоритмічно важка задача. Є більш ефективні алгоритми.

Алгоритм Краскала (пошук остового дерева мінімальної ваги у зв'язному графі)

Нехай є граф G , $|V| = n$ та граф N_n (порожній граф, вершини якого збігаються з вершинами G). Будуємо граф $T_1 = N_n \cup \{e_1\}$, де e_1 – ребро графа G , що має найменшу вагу (якщо таких ребер декілька, вибираємо будь-яке з них). На кожному наступному кроці приєднуємо до вже побудованого графа T_i ребро e_{i+1} графа G , яке: 1) відмінне від всіх попередньо приєднаних ребер; 2) має найменшу вагу серед тих ребер, що залишились; 3) не утворює циклів із ребрами графа T_i . Якщо граф T_i побудований та $i < n-1$, то T_i не зв'язний. Отже, при $i < n-1$ граф T_{i+1} завжди можна побудувати. Алгоритм припиняє роботу, коли буде побудовано граф T_{n-1} . Цей граф і є шуканим остовним деревом мінімальної ваги.

Зауваження. Алгоритм Прима відрізняється від алгоритму Краскала лише тим, що на кожному кроці будується піддерево шуканого остового дерева.

Алгоритм Дейкстри знаходження найкоротшого шляху між парою вершин у навантаженому оргграфі

Дано зважений орієнтований граф. Виділено дві його вершини. Треба знайти найкоротший шлях від однієї з них до іншої. Безпосереднє порівняння

довжин всіх можливих шляхів з метою знаходження найкоротшого при великій кількості вершин (а це як правило має місце в реальній задачі) стає неефективним. Тому створено кілька алгоритмів пошуку найкоротшого шляху.

В алгоритмі Дейкстри вершинам приписуються мітки – тимчасові або постійні. Результати записують у таблицю. Елементи рядка (тимчасові мітки) для вершини u , яка останньою отримала постійну мітку, перераховуються за правилом

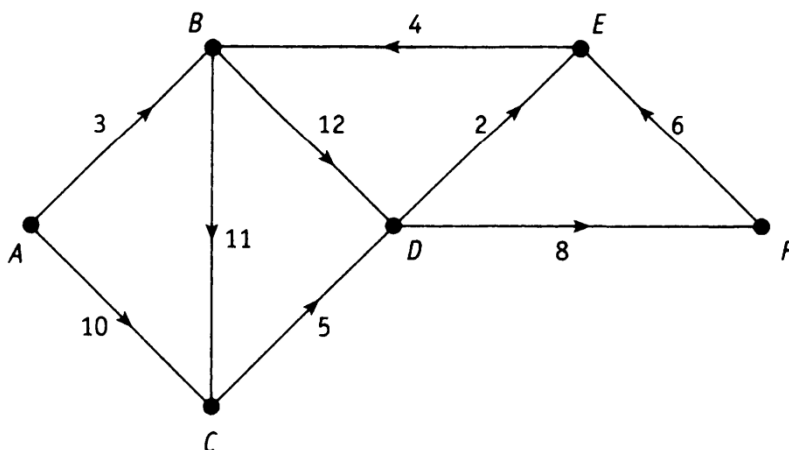
$$d(v) = \min \{ \text{стара мітка}, \text{остання нова мітка } u + \text{вага дуги } (u, v) \}.$$

Алгоритм Дейкстри закінчує роботу, коли всі вершини отримають постійні мітки. З таблиці легко отримати найкоротший шлях і послідовність вершин цього шляху.

Зауваження. Модифікований алгоритм дає найкоротші шляхи від фіксованої вершини до всіх інших вершин.

Приклад оформлення розв'язання «вручну» задачі на знаходження найкоротшого шляху

Дано зважений орієнтований граф. Знайти найкоротший (A, F) – шлях.



Розв'язання.

Перший рядок таблиці заповнюється на **нульовому кроці** алгоритму вагою дуги (A, x) , якщо така існує, і знаком ∞ у протилежному випадку

№ кроку	Вершини з постійними мітками	Відстань до вершини						Вершини з тимчасовими мітками
		A	B	C	D	E	F	
0	A	0	3	10	∞	∞	∞	B, C, D, E, F
1	B	0	3	10	15	∞	∞	C, D, E, F
2	C	0	3	10	15	∞	∞	D, E, F
3	D	0	3	10	15	17	23	E, F
4	E	0	3	10	15	17	23	

Перший крок

- знаходимо в першому рядку мінімальне число, воно дорівнює 3,
 - відмічаємо його (напівжирний шрифт) і відзначаємо, що вершина B отримала постійну мітку 3,
 - записуємо B в другий стовпчик, а решту вершин в останній стовпчик,
 - перераховуємо тимчасові мітки за формулою $d(v) = \min\{\text{стара мітка, остання нова мітка } u + \text{вага дуги } (u, v)\}$
- (заповнюємо другий рядок таблиці):

$$l(C) := \min\{10, 3 + 11\} = 10,$$

$$l(D) := \min\{\infty, 3 + 12\} = 15,$$

$$l(E) := \min\{\infty, 3 + \infty\} = \infty,$$

$$l(F) := \min\{\infty, 3 + \infty\} = \infty,$$

Другий крок

- знаходимо мінімальну мітку 10 і відмічаємо вершину C ,
- перераховуємо тимчасові мітки (заповнюємо третій рядок таблиці):

$$l(D) := \min\{15, 10 + 5\} = 15,$$

$$l(E) := \min\{\infty, 10 + \infty\} = \infty,$$

$$l(F) := \min\{\infty, 10 + \infty\} = \infty,$$

Третій крок

- знаходимо мінімальну мітку 15 і відмічаємо вершину D ,
- перераховуємо тимчасові мітки (заповнюємо четвертий рядок таблиці):

$$l(E) := \min\{\infty, 15 + 2\} = 17,$$

$$l(F) := \min\{\infty, 15 + 8\} = 23,$$

Четвертий крок

- знаходимо мінімальну мітку 17 і відмічаємо вершину E ,
- перераховуємо тимчасові мітки (заповнюємо п'ятий рядок таблиці):

$$l(F) := \min\{23, 17 + \infty\} = 23.$$

№ кроку	Відмічені вершини	Відстань до вершини						Невідмічені вершини
		A	B	C	D	E	F	
0	A	0	3	10	∞	∞	∞	B, C, D, E, F
1	B	0	3	10	15	∞	∞	C, D, E, F
2	C	0	3	10	15	∞	∞	D, E, F
3	D	0	3	10	15	17	23	E, F
4	E	0	3	10	15	17	23	

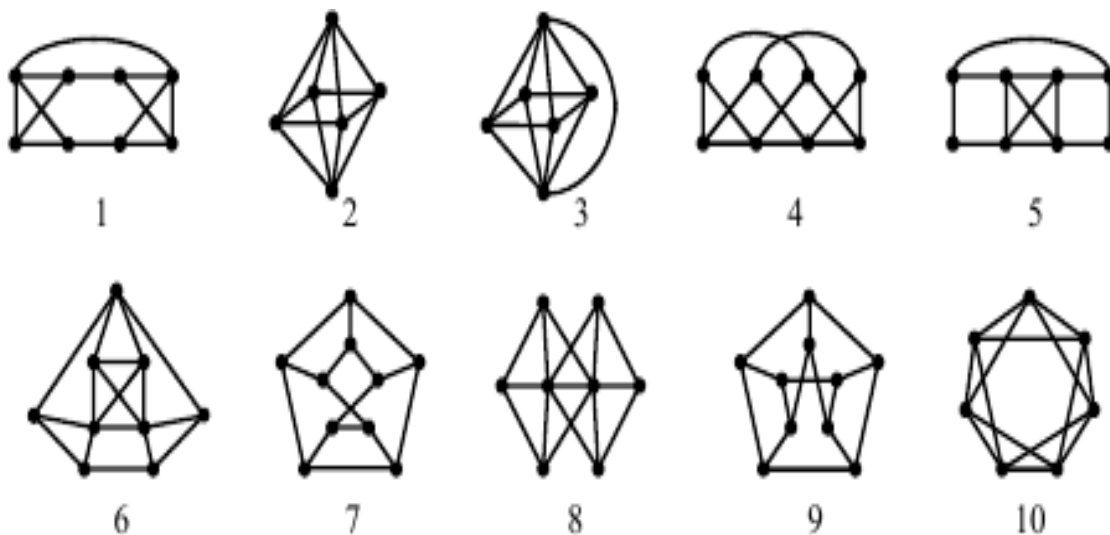
Висновки:

- довжина найкоротшого (A, F) – шляху дорівнює 23,

2) для запису шляху користуються таким правилом: останньою є вершина F – кінцева. Свою постійну мітку вона «отримала з вершини D », тому передостанньою вершиною найкоротшого шляху буде саме D . Вершина D свою постійну мітку 15 отримала з вершини B , а вершина A є початком шляху. Отже, найкоротший (A,F) – шлях проходить через вершини A,B,D,F .

Завдання лабораторної роботи

Завдання 1. Занумерувати вершини графа. В якості ваги ребер використати числа з набору 1,1,2,2,2,3,3,4,4,5,6,7,7,10. Застосувати алгоритм Краскала побудови мінімального остовного дерева. Знайти центр дерева й визначити пару найбільш віддалених від центра вершин, позначити їх A та B (10 варіантів).



Завдання 2. Застосувати алгоритм Прима для знаходження мінімального остовного дерева графа, починаючи з вказаної вершини (рис 7 для варіантів 1–8, рис. 8 – для варіантів 6 - 10):

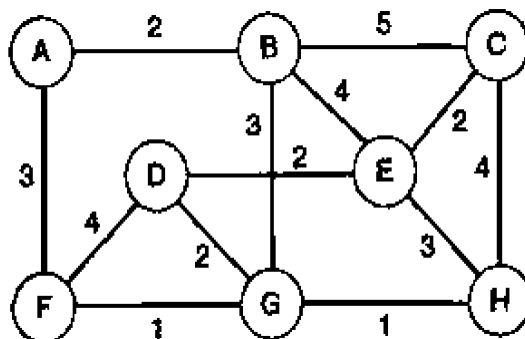


Рисунок 7 – Для завдання 2

- 1) починаючи з вершини A ,
- 2) починаючи з вершини B ,
- 3) починаючи з вершини C ,

- 4) починаючи з вершини D,
- 5) починаючи з вершини E.

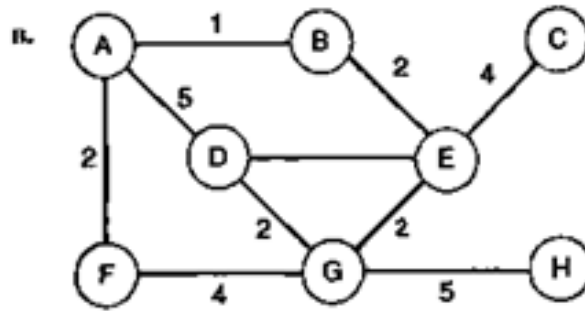


Рисунок 8 – Для завдання 2

- 6) починаючи з вершини A,
- 7) починаючи з вершини B,
- 8) починаючи з вершини F,
- 9) починаючи з вершини G,
- 10) починаючи з вершини H.

Завдання 3. Дано зважений орієнтований граф. Застосувати алгоритм Дейкстри для знаходження найкоротшого шляху

Варіант 1 – від вершини x_0 до вершини x_8 на рис. 9;

Варіант 2 – від вершини x_0 до вершини x_6 на рис. 9;

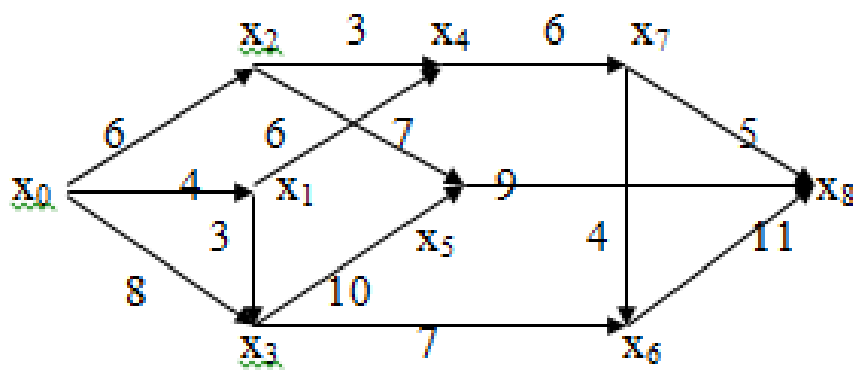


Рисунок 9 – До завдання 3

Варіант 3 – від вершини v_1 до вершини v_7 на рис. 10;

Варіант 4 – від вершини v_2 до вершини v_7 на рис. 10;

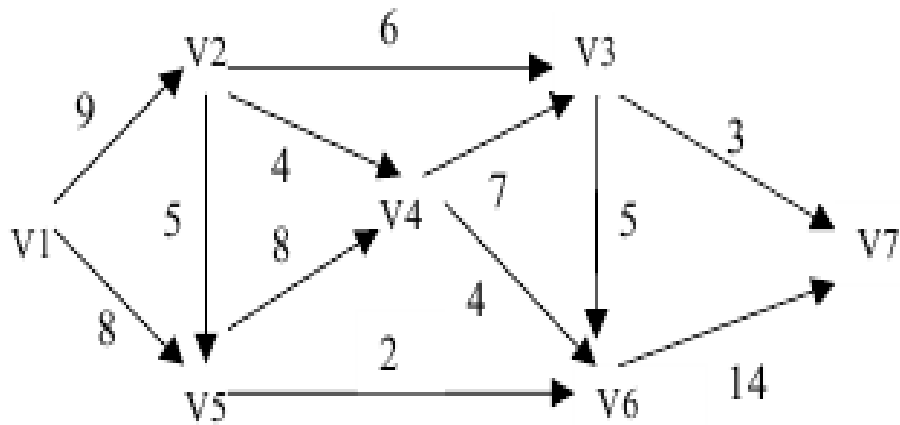


Рисунок 10 – До завдання 3

Варіант 5 – від вершини S до вершини T (орієнтація зліва направо й знизу вверху) на рис. 11;

Варіант 6 – від вершини A до вершини T (орієнтація зліва направо й знизу вверху) на рис. 11;

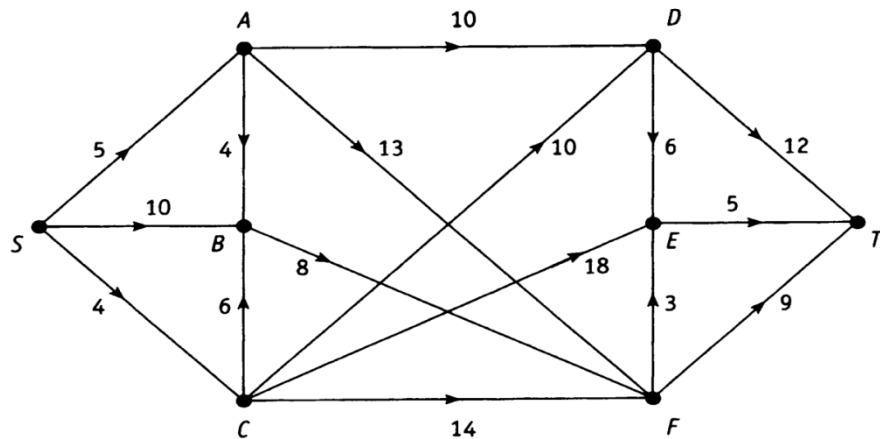


Рисунок 11 – До завдання 3

Варіант 7 – від вершини 1 до вершини 11 (орієнтація зліва направо й знизу вверху) на рис. 12;

Варіант 8 – від вершини 3 до вершини 11 (орієнтація зліва направо й знизу вверху) на рис. 12;

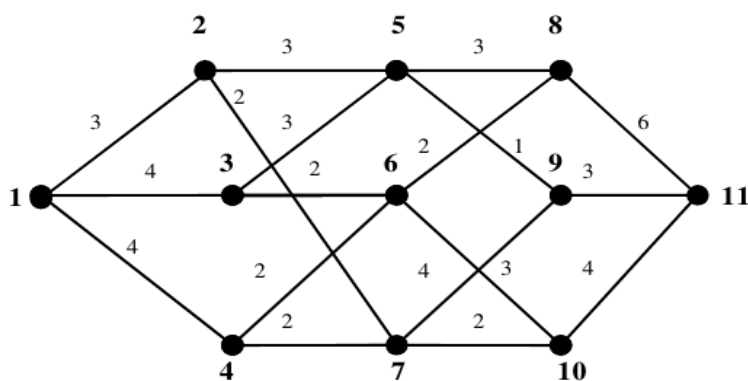


Рисунок 12 – До завдання 3

Варіант 9 – від вершини А до вершини Н (орієнтація зліва направо й знизу вверху) на рис. 13;

Варіант 10 – від вершини F до вершини С (орієнтація зліва направо й знизу вверху) на рис. 13.

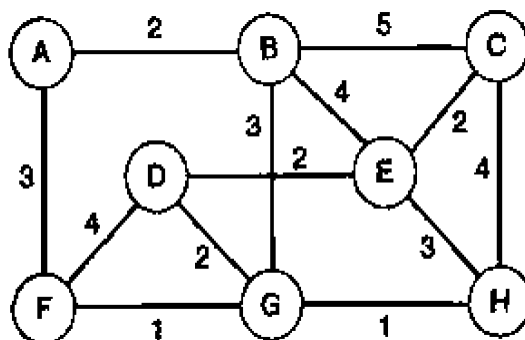


Рисунок 13 – До завдання 3

Лабораторне заняття 14 Знаходження максимального потоку в мережі

Мета. Ознайомлення з поняттями мережі, потоку, розрізу, теоремою Форда-Фалкерсона. Розв'язання задачі про максимальний потік методом Форда-Фалкерсона вручну й за допомогою відповідної комп'ютерної програми.

Необхідне обладнання та матеріали.

Лекція 7

Бондаренко М. Ф. Комп'ютерна дискретна математика : підручник для студ. вузів. Харків : Компанія СМІТ, 2004. 445 с.

Журавчак Л. М. Дискретна математика для програмістів : навч. посіб. Львів : Львівська політехніка, 2019. 420 с.

Актуалізація знань.

♫ **Увага!** Дискусія з питань:

- Поняття потоку, розрізу, максимального потоку в мережі;
- Формулювання й використання теореми Форда-Фалкерсона;
- Описання кроків алгоритму знаходження максимального потоку.

Хід виконання роботи.



Розглядається задача максимізації потоку деякого продукту по мережі. Подібного роду задачі виникають при організації перекачування нафти або газу по трубопроводах, при плануванні залізничного або автомобільного руху, передачі інформації по мережам тощо.

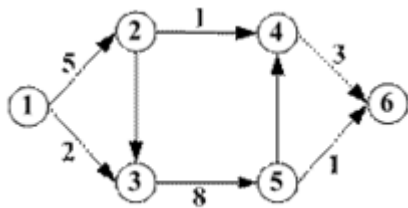
Наведемо необхідні теоретичні відомості, які допоможуть формалізувати вказані задачі.

Означення. *Мережею* називають зважений орієнтований граф з одним витокком (з цієї вершини дуги тільки виходять) і одним стоком (в цю вершину дуги тільки входять). Вага дуги в мережі називається її *пропускною спроможністю*.

Приклади вершин мережі: перехрестя доріг, телефонні вузли, залізничні вузли, аеропорти, склади і т.і.

Приклади дуг мережі: дороги, труби, телефонні або залізничні лінії і т.і.

Приклад транспортної мережі:

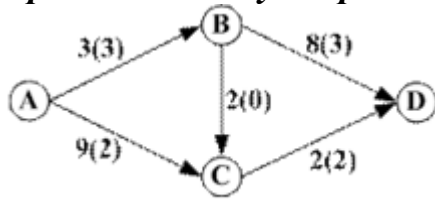


Означення. *Потоком* в транспортній мережі називається невід'ємна функція, визначена на множині дуг мережі, яка задовольняє двом умовам:

- 1) величина потоку по кожній дузі не перевершує її пропускної спроможності;
- 2) сума потоків, що входять в кожну вершину мережі, за винятком витокку та стоку, дорівнює сумі потоків, що виходять з вершини.

Означення. *Величиною потоку* є сума потоків, що виходять з витокку або сума потоків, що входять в сток мережі.

Приклад потоку в транспортній мережі:

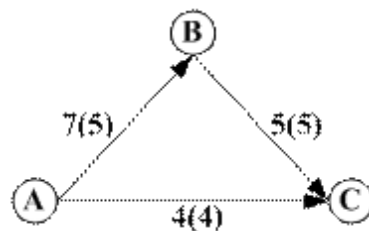


Числа в дужках – потоки на дугах, без дужок – пропускні спроможності, A – виток, D – сток, величина потоку: $3+2=5$.

Означення. Розрізом транспортної мережі називається така множина дуг, видалення яких відділяє витік від стоку.

Означення. Мінімальним розрізом транспортної мережі називається розріз з мінімальною пропускною спроможністю.

Приклад. Мережа



має два розрізи $\{(A,B),(A,C)\}$ і $\{(A,C),(B,C)\}$. Пропускна спроможність першого розрізу дорівнює $11 = 7+4$, а другого – $9 = 4+5$, тому другий розріз є мінімальним.

Алгоритм знаходження максимального потоку в мережі заснований на такій теоремі.

Теорема (Форда – Фалкерсона). Величина максимального потоку в мережі дорівнює величині мінімального розрізу.

В наведеному прикладі максимальний потік в мережі дорівнює $9 = \min(11, 9)$. Цей максимальний потік вказаний в круглих дужках.

Розглянемо ланцюги мережі, які сполучають витік зі стоком.

Означення. Дуга ланцюга називається допустимою дугою, якщо:

- 1) напрямок дуги співпадає з напрямком потоку і потік по цій дузі менший за її пропускну спроможність;
- 2) напрямок дуги протилежний напрямку потоку і потік по цій дузі більше нуля.

Означення. Ланцюг, що з'єднує витік мережі зі стоком, називається збільшуючим, якщо всі його дуги є допустимими.

Алгоритм Форда-Фалкерсона знаходження максимального потоку в мережі

1 крок. Якщо потік в мережі не заданий, то вважають потік нульовим.

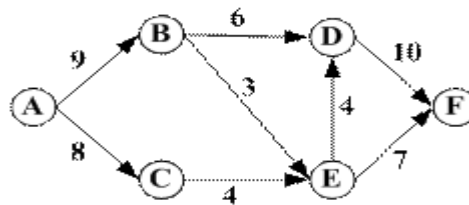
2 крок. Виконати дії:

- взяти будь-який збільшуючий ланцюг;

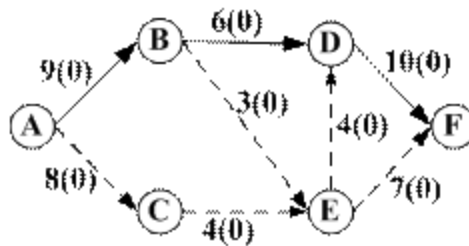
- обчислити найменшу різницю d між пропускними спроможностями дуг цього ланцюга і потоками по цим дугам;
- потоки по дугам, напрямок яких співпадає з напрямком потоку, збільшити на d ;
- потоки по дугам, напрямок яких протилежний напрямку потоку, зменшити на d .

3 крок. Якщо в мережі є збільшуючий ланцюг, то переходимо до кроку 2. В противному випадку максимальний потік побудований.

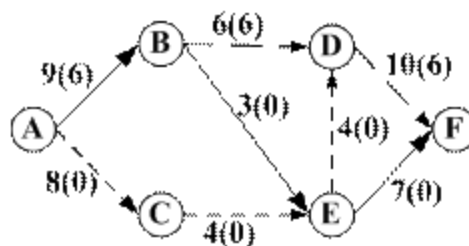
Задача 1. Побудувати максимальний потік для заданої транспортної мережі.



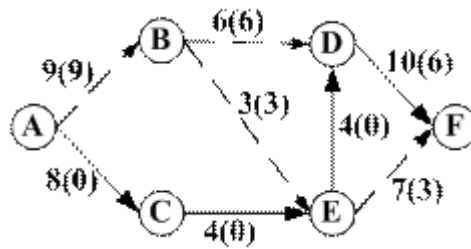
Розв'язання. Потік в мережі не заданий, будемо вважати його нульовим.



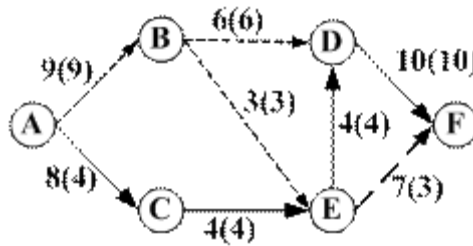
Розглянемо збільшуючий ланцюг AB, BD, DF . Напрямок всіх дуг співпадає з напрямком потоку, $d = \min(9-0, 6-0, 10-0) = 6$. Нові потоки по дугам цього ланцюга: $AB: 0+6=6$, $BD: 0+6=6$, $DF: 0+6=6$.



Є ще збільшуючий ланцюг: AB, BE, EF . Напрямок дуг співпадає з напрямком потоку, $d = \min(9-6, 3-0, 7-0) = 3$. Нові потоки по дугам ланцюга: $AB: 6+3=9$, $BE: 0+3=3$, $EF: 0+3=3$.

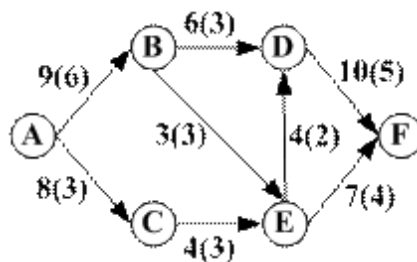


Знаходимо ще один збільшуючий ланцюг: AC, CE, ED, DF . Напрямок дуг співпадає з напрямком потоку, $d = \min(8-0, 4-0, 4-0, 10-6) = 4$. Нові потоки по дугам ланцюга: $AC: 0+4=4$, $CE: 0+4=4$, $ED: 0+4=4$, $DF: 6+4=10$.

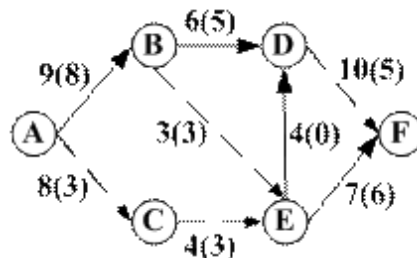


Більше збільшуючих ланцюгів в мережі немає, тому максимальний потік побудований і дорівнює $13=9+4=10+3$.

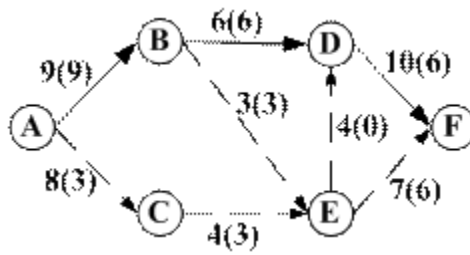
Задача 2. Побудувати максимальний потік для заданої транспортної мережі (потік в мережі заданий).



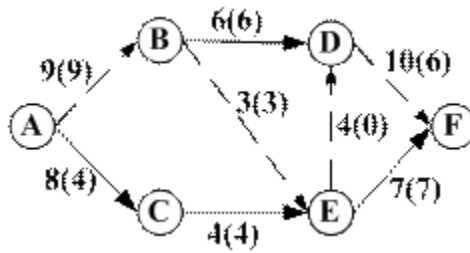
Розв'язання. Збільшуючий ланцюг: AB, BD, DE, EF . Напрямок дуги DE протилежний напрямку потоку, напрямки інших дуг співпадає з напрямком потоку, $d = \min(9-6, 6-3, 4-2, 7-4) = 2$. Нові потоки по дугам ланцюга: $AB: 6+2=8$, $BD: 3+2=5$, $DE: 2-2=0$, $EF: 4+2=6$



У збільшуючому ланцюзі AB, BD, DF напрямки дуг співпадає з напрямком потоку, $d = \min(9-8, 6-5, 10-5) = 1$. Нові потоки по дугам ланцюга: $AB: 8+1=9$, $BD: 5+1=6$, $DF: 5+1=6$.



Є ще один збільшувачий ланцюг: AC, CE, EF . Напрямок дуг співпадає з напрямком потоку, $d = \min(8-3, 4-3, 7-6) = 1$. Нові потоки по дугам ланцюга: $AC: 3+1=4$, $CE: 3+1=4$, $EF: 6+1=7$.



Збільшувачих ланцюгів більше в мережі немає, тому максимальний потік побудований і дорівнює $13=9+4=10+3$.

Завдання лабораторної роботи

Завдання 1. Знайти максимальний потік в мережі

Варіант 1 – на рис.14;

Варіант 2 – на рис. 14, помінявши місцями пропускні спроможності ребер AB та BF ;

Варіант 3 – на рис. 14, помінявши місцями пропускні спроможності ребер CF та FT ;

Варіант 4 – на рис. 14, помінявши місцями пропускні спроможності ребер CD та CB ;

Варіант 5 – на рис. 14, помінявши місцями пропускні спроможності ребер AB та DE ;

Варіант 6 – на рис. 14, помінявши місцями пропускні спроможності ребер SB та ET ;

Варіант 7 – на рис. 14, помінявши місцями пропускні спроможності ребер SB та ET ;

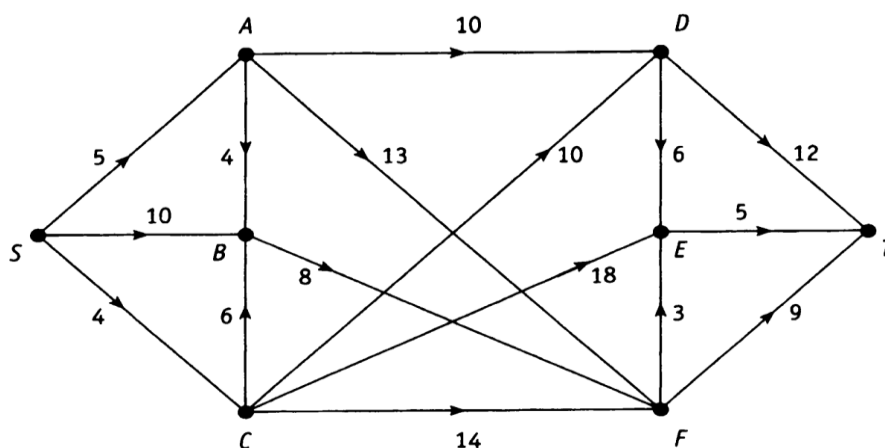


Рисунок 14 – До завдання 1

Варіант 8 – на рисунку 15;

Варіант 9 – на рис. 15, помінявши місцями пропускні спроможності ребер x_1x_4 та x_3x_5 ;

Варіант 10 – на рис. 15, помінявши місцями пропускні спроможності ребер x_1x_3 та x_7x_6 .

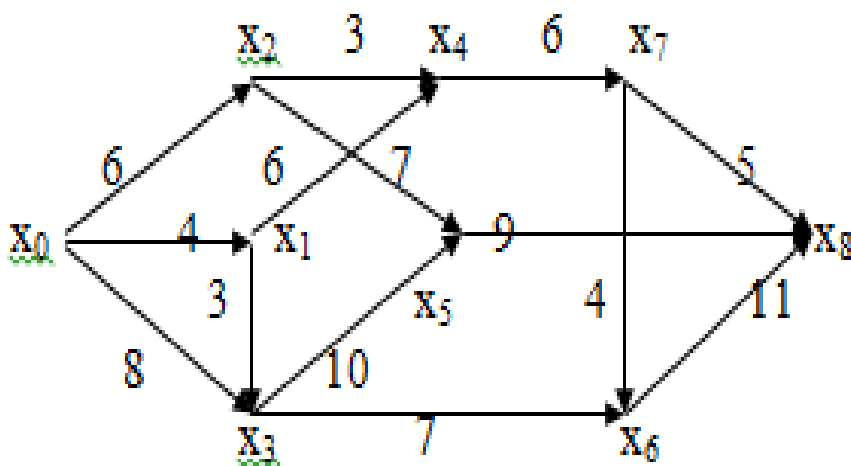


Рисунок 15 – До завдання 1

Лабораторне заняття 15 *Розв'язання задачі про призначення*

Мета. Розв'язання задачі про призначення «вручну» й венгерським методом за допомогою відповідної комп'ютерної програми. Тестування програми.

Необхідне обладнання та матеріали.

Лекція 7

Бондаренко М. Ф. Комп'ютерна дискретна математика : підручник для студ. вузів. Харків : Компанія СМІТ, 2004. 445 с.

Журавчак Л. М. Дискретна математика для програмістів : навч. посіб. Львів : Львівська політехніка, 2019. 420 с.

Спекторський І. Я. Дискретна математика. Збірник задач : навчальний посібник. URL :

http://spectorsky.ho.ua/files/rasch_%D1%81%D0%B0%D0%BC%D0%A0%D0%BE%D0%B1.pdf

Актуалізація знань.

📖 **Увага!** Дискусія з питань:

- Сутність задачі про призначення;
- Аналоги задачі про призначення;
- Матричне задання задачі про призначення;
- Відмінність задач про призначення на мінімум і на максимум.

Хід виконання роботи.



Задача про призначення – одна з фундаментальних задач комбінаторної оптимізації. Задача полягає в пошуку мінімальної суми дуг у зваженому орієнтованому дводольному графі.

Якщо множина V вершин простого графа допускає розбиття на дві підмножини V_1 та V_2 ($V_1 \cap V_2 = \emptyset$) так, що не існує ребер, що поєднують вершини однієї і тієї ж підмножини, то граф називається дводольним або біграфом (рис. 16).

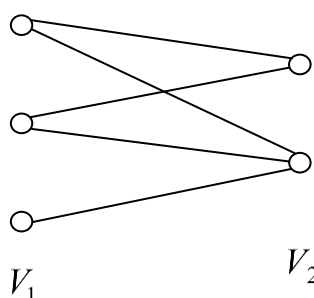


Рисунок 16 – Дводольний граф

Теорема 6.4 (критерій Кеніга дводольності графа). Зв'язний граф G є дводольним тоді й тільки тоді, коли він не має циклів непарної довжини.

У найбільш загальній формі задача про призначення формулюється наступним чином:

Маємо деяку кількість робіт та деяку кількість виконавців. Будь-який виконавець може бути призначений для виконання будь-якої (але тільки однієї)

роботи, але з різними витратами. Треба розподілити роботи так, щоб виконати всі види робіт з мінімальними затратами.

Якщо число робіт співпадає з числом виконувачів, то задача називається лінійною задачею про призначення. Зазвичай, якщо говорять про задачу про призначення без додаткових умов, мають на увазі саме лінійну задачу про призначення.

Приклад. Припустимо, що таксомоторна компанія має три вільні машини (виконувачі), й три зазамовника (роботи), які бажають отримати таксі якнайшвидше. Фірма турбується про час доставки таксі до замовника, тому для кожної машини *вартість визначається* кількістю часу, протягом якого машина добереться до місця, визначеного замовником.

Розв'язком задачі буде такий розподіл машин по замовникам, що сумарна вартість (сумарний час чекання) буде мінімальним.

Зауваження. Задачу про призначення можна зробити більш гнучкою. У наведеному прикладі може виявитися не три, а чотири вільні таксі, а замовників як і раніше три. Можна призначити четвертого (фіктивного) замовника з нульовою вартістю, розподіл машини на фіктивного замовника означає— «нічого не роби».

Аналогічний прийом можна використовувати у випадку, коли число замовників перевищує число доступних машин, й інші варіанти, наприклад, поставити задачу збільшення доходу, а не мінімізацію витрат.

Розв'язання задачі про призначення венгерським методом. Випадок задачі на максимум

Наприклад, «На роботу влаштовуються 6 кандидатів на 6 вакансій і вони отримали відповідні оцінки на співбесіді на кожену вакансію. Провести набір кандидатів так, щоб сумарна оцінка кандидатів була максимальною».

Будемо вважати, що дані занесено в матрицю. Треба вибрати по одній клітці в рядку і стовпчику так, щоб їх сума була *максимальною*.

$$\begin{pmatrix} 6 & 15 & 3 & 12 & 4 & 2 \\ 14 & 3 & 3 & 7 & 2 & 1 \\ 3 & 2 & 8 & 15 & 8 & 12 \\ 3 & 14 & 3 & 15 & 11 & 10 \\ 3 & 13 & 1 & 9 & 6 & 6 \\ 15 & 10 & 3 & 4 & 5 & 10 \end{pmatrix}$$

Розв'язання.

Крок 1:

Зауваження: перший крок буде тільки для задачі на максимум. Якщо треба розв'язати задачу на мінімум, то він пропускається.

Перетворимо матрицю, замінивши кожний елемент кожного рядка різницею максимального елемента цього рядка й самого елемента.

$$\left(\begin{array}{cccccc|l} 6 & 15 & 3 & 12 & 4 & 2 & \max = 15 \\ 14 & 3 & 3 & 7 & 2 & 1 & \max = 14 \\ 3 & 2 & 8 & 15 & 8 & 12 & \max = 15 \\ 3 & 14 & 3 & 15 & 11 & 10 & \max = 15 \\ 3 & 13 & 1 & 9 & 6 & 6 & \max = 13 \\ 15 & 10 & 3 & 4 & 5 & 10 & \max = 15 \end{array} \right)$$

Отримаємо

$$\left(\begin{array}{cccccc} 9 & 0 & 12 & 3 & 11 & 13 \\ 0 & 11 & 11 & 7 & 12 & 13 \\ 12 & 13 & 7 & 0 & 7 & 3 \\ 12 & 1 & 12 & 0 & 4 & 5 \\ 10 & 0 & 12 & 4 & 7 & 7 \\ 0 & 5 & 12 & 11 & 10 & 5 \end{array} \right)$$

Крок 2.

Треба отримати нулі в кожному рядку й в кожному стовпці. В третьому, п'ятому й шостому стовпцях нулів немає. Віднімемо з елементів цих столпців мінімальний елемент відповідного стовпця.

9	0	12	3	11	13
0	11	11	7	12	13
12	13	7	0	7	3
12	1	12	0	4	5
10	0	12	4	7	7
0	5	12	11	10	5
min = 7			min = 4	min = 3	

$$\left(\begin{array}{cccccc} 9 & 0 & 5 & 3 & 7 & 10 \\ 0 & 11 & 4 & 7 & 8 & 10 \\ 12 & 13 & 0 & 0 & 3 & 0 \\ 12 & 1 & 5 & 0 & 0 & 2 \\ 10 & 0 & 5 & 4 & 3 & 4 \\ 0 & 5 & 5 & 11 & 6 & 2 \end{array} \right)$$

Крок 3.

Отримали матрицю, в якій в кожному рядку й кожному стовпці є нуль. Нам треба відмітити по одному нулю в кожному рядку й кожному стовпці. В цій матриці тільки перші чотири рядки й стовпці задовольняють вимозі. Відмітимо відповідні комірки рамкою. Відмітимо 5-й рядок як «незадоволений», в ньому ми такий нуль відмітити не змогли, й другий

стовпчик, він містить нуль в п'ятому рядку. Але другий стовпчик також містить нуль у першому рядку. Відмітимо і його як «незадоволений». Перший рядок нулів більше не має, тобто процес відмічання «незадоволених» рядків завершено. Ми отримали ситуацію під назвою «вузьке місце». В таблиці будемо відмічати «незадоволені» рядки й стовпці зірочками, а число поруч із зірочкою буде означати порядок відмічання (для кращого розуміння процесу).

9	0	5	3	7	10	* (3)
0	11	4	7	8	10	
12	13	0	0	3	0	
12	1	5	0	0	2	
10	0	5	4	3	4	* (1)
0	5	5	11	6	2	
* (2)						

Виберемо мінімальний елемент в помічених (*) рядках поза рядків і стовпців з відміченими нулями. Це число 3 в п'ятому стовпці й п'ятому рядку. Віднімемо цей елемент з відмічених рядків й додамо до відмічених стовпців.

9	0	5	3	7	10	-3
0	11	4	7	8	10	
12	13	0	0	3	0	
12	1	5	0	0	2	
10	0	5	4	3	4	-3
0	5	5	11	6	2	
+3						

6	0	2	0	4	7
0	14	4	7	8	10
12	16	0	0	3	0
12	4	5	0	0	2
7	0	2	1	0	1
0	8	5	11	6	2

Зауважимо, що тепер можна відмітити нуль в п'ятому рядку й п'ятому стовпці.

Крок 4.

Ще немає нуля в 6-ому рядку. відмітимо його як незадоволений, він має нуль в першому стовпці, відмітимо його як незадоволений. Він, в свою чергу, містить нуль у другому рядку, відмітимо його, але в ньому нулів більше немає. Процес відмічання завершено.

$$\begin{array}{cccccc|c}
 6 & \boxed{0} & 2 & 0 & 4 & 7 & \\
 \boxed{0} & 14 & 4 & 7 & 8 & 10 & *(3) \\
 12 & 16 & \boxed{0} & 0 & 3 & 0 & \\
 12 & 4 & 5 & \boxed{0} & 0 & 2 & \\
 7 & 0 & 2 & 1 & \boxed{0} & 1 & \\
 0 & 8 & 5 & 11 & 6 & 2 & *(1) \\
 \hline
 *(2) & & & & & &
 \end{array}$$

Виберемо мінімальний елемент у відмічених рядках поза відмічених стовпчиках. Це елемент 2 в шостому рядку і шостому стовпчику. Віднімемо двійку з другого та шостого рядків і додамо допершого стовпчика.

$$\begin{array}{cccccc|c}
 6 & \boxed{0} & 2 & 0 & 4 & 7 & \\
 \boxed{0} & 14 & 4 & 7 & 8 & 10 & -2 \\
 12 & 16 & \boxed{0} & 0 & 3 & 0 & \\
 12 & 4 & 5 & \boxed{0} & 0 & 2 & \\
 7 & 0 & 2 & 1 & \boxed{0} & 1 & \\
 0 & 8 & 5 & 11 & 6 & 2 & -2 \\
 \hline
 +2 & & & & & &
 \end{array}$$

Тепер можна відмітити ще один нуль.

$$\begin{pmatrix}
 8 & \boxed{0} & 2 & 0 & 4 & 7 \\
 \boxed{0} & 12 & 2 & 5 & 6 & 8 \\
 14 & 16 & \boxed{0} & 0 & 3 & 0 \\
 14 & 4 & 5 & \boxed{0} & 0 & 2 \\
 9 & 0 & 2 & 1 & \boxed{0} & 1 \\
 0 & 6 & 3 & 9 & 4 & \boxed{0}
 \end{pmatrix}$$

Отримали матрицю з шістьма нулями, по одному в кожному рядку й кожному стовпці. Отже, можна провести призначення (розподіл робіт тощо) за матрицею:

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

$$\begin{pmatrix}
 6 & 15 & 3 & 12 & 4 & 2 \\
 14 & 3 & 3 & 7 & 2 & 1 \\
 3 & 2 & 8 & 15 & 8 & 12 \\
 3 & 14 & 3 & 15 & 11 & 10 \\
 3 & 13 & 1 & 9 & 6 & 6 \\
 15 & 10 & 3 & 4 & 5 & 10
 \end{pmatrix}$$

Вартість такого призначення дорівнює: $15+14+8+15+6+10=68$.

Завдання лабораторної роботи

Завдання 1. Розв'язати задачу про призначення із заданою ваговою матрицею (n – номер варіанта)

$$\begin{pmatrix} 3 & 2 & 4 & 2 \\ n & 4 & 2 & 3 \\ 7 & 2 & 4 & 23-n \\ 2 & 2 & 3 & 4 \end{pmatrix}$$

ВИКОРИСТАНА ЛІТЕРАТУРА

1. Бондаренко М. Ф. Комп'ютерна дискретна математика : підручник для студ. вузів. Харків : Компанія СМІТ, 2004. 445 с.
2. Борисенко О. А. Дискретна математика : підручник для студентів вищих навчальних закладів. Суми : Університетська книга, 2019. 255 с.
3. Журавчак Л. М. Дискретна математика для програмістів : навч. посіб. Львів : Львівська політехніка, 2019. 420 с.
4. Капітонова Ю. В. Основи дискретної математики : підручник. Київ : Наукова думка, 2012. 579 с.
5. Стеганцева П. Г., Гречнева М. О, Манько Н. І., Спиця О. Г., Стеганцев Є. В. Дискретна математика : навчальний посібник для здобувачів ступеня бакалавра освітньо-професійних програм «Середня освіта (математика)», «Середня освіта (інформатика)», «Математика», «Комп'ютерна математика», «Комп'ютерне моделювання», «ІСТ». Запоріжжя : ЗНУ, 2021. 178 с.
6. Стеганцева П. Г., Гречнева М., Стеганцев Є. В. Математична логіка : навчальний посібник для здобувачів ступеня вищої освіти бакалавра спеціальності «Математика» освітньо-професійних програм «Математика», «Комп'ютерна математика». Запоріжжя : ЗНУ, 2020. 103 с.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Бардачов Ю. М. Дискретна математика : підруч. для студ. вищ. техн. навч. закл. Київ : Вища школа, 2007. 548 с.
2. Скрябіна А. В., Стеганцева П. Г. Перерахування усіх То-топологій, заданих одним вектором, та підрахунок То-топологій певної ваги. Матеріали XXIII Міжнародного науково-практичного семінару ім. А.Я.Петренюка, присвяченому 70-річчю Льотної академії Національного авіаційного університету «Комбінаторні конфігурації та їхні застосування». Запоріжжя-Кропивницький : ПП Ексклюзив-Систем, 2021. С. 165-168.
3. Стеганцева П. Г., Артеменко А. О. Рекурентні співвідношення для числа неізоморфних (n,m) - графів. *Вісник Запорізького національного університету. Фізико-математичні науки*. 2021. № 1. С. 51-56. URL: <https://doi.org/10.26661/2413-6549-2021-1-06>.
4. Стеганцева П. Г., Скрябіна А. В. Топології на n -елементній множині, узгоджені з топологіями близькими до дискретних на $(n-1)$ -елементній множині. *Український математичний журнал*. 2021. Т. 73. № 2. С. 238-248. URL: <http://umj.imath.kiev.ua/index.php/umj/article/view/6174>.
5. Choporov, S., Homeniuk, S., Grebenyuk, S. Optimized smoothing of discrete models of the implicitly defined Geometrical Objects' surfaces. *Eastern-European Journal of Enterprise Technologi*~~this link is disabled~~. 2018, 3(4-93). P. 52–60.
6. Skryabina A. V., Stegantseva P. G. The relation between To-topologies whis the weight k on n -element set and To-topologies close to the discrete on $(n-1)$ -element set. Algebraic and Geometric Methods of Analysis May 25-28, 2021 Odesa, Ukraine dedicate to the memory of Yuriy Trokhymchuk (17.03.1928-18.12.2019) Одеса : ONAFT, 2021. С. 142. URL : <https://www.imath.kiev.ua/~topology/conf/agma2021/contents/agma2021-abstracts.pdf>.

ІНФОРМАЦІЙНІ РЕСУРСИ

1. Базилевич Л. Дискретна математика у прикладах і задачах : підручник. URL : http://chyslo.com.ua/index.php?route=product/product&product_id=57
2. Коцовський В. М. Основи дискретної математики : навчальний посібник. URL : <https://dspace.uzhnu.edu.ua/jspui/handle/lib/31664>
3. Спекторський І. Я. Дискретна математика. Збірник задач : навчальний посібник. URL : <https://ela.kpi.ua/handle/123456789/11562>

Навчальне видання
(українською мовою)

Стеганцева Поліна Георгіївна
Стеганцев Євгеній Вікторович

КОМП'ЮТЕРНА ДИСКРЕТНА МАТЕМАТИКА І АНАЛІЗ ДАНИХ

Методичні рекомендації
до лабораторних занять
для здобувачів ступеня вищої освіти бакалавра
спеціальності «Математика»
освітньо-професійної програми «Математика»

Рецензент *Є.В. Панасенко*
Відповідальний за випуск *І.В. Зіновєєв*
Коректор *Є.В. Стеганцев*