

ДМіністерство освіти і науки України
Національний університет «Запорізька Політехніка»

Кафедра програмних засобів

ЗВІТ

з лабораторної роботи №3

з дисципліни «Комп'ютерна дискретна математика»

На тему «ОСНОВНІ ОПЕРАЦІЇ НАД ГРАФАМИ»

Варіант №20

Виконав:

Студент групи КНТ-122

О. А. Онищенко

Прийняли:

Ст. Викладач

О. А. Щербина

2023

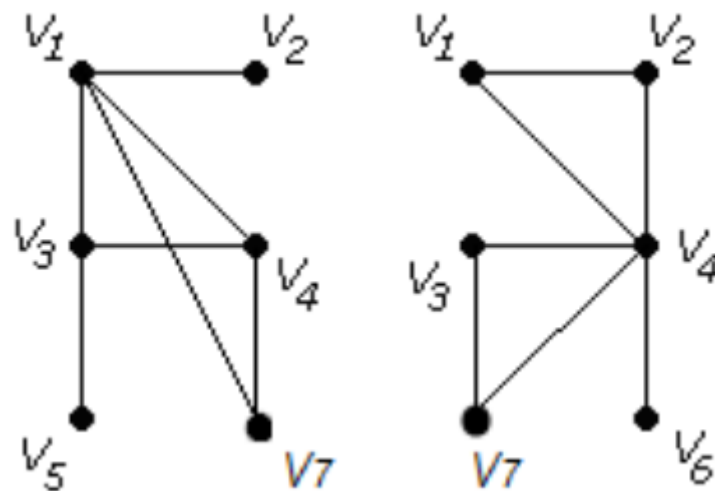
Текст завдання №1	3
Розв'язок завдання №1	5
Текст завдання №2	9
Розв'язок завдання №2	9
Результати роботи завдання №2	11
Блок-схема завдання №2	12

Текст завдання №1

1. Виконати наступні операції над графами:

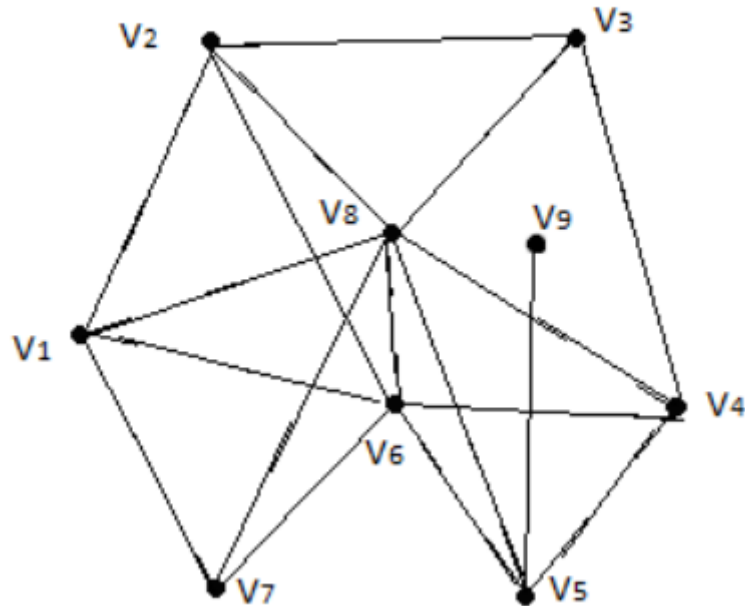
- знайти доповнення до першого графу,
- об'єднання графів,
- кільцеву суму $G1 \oplus G2$,
- розщепити вершину у другому графі,
- виділити підграф A , що складається з 3-х вершин в $G1$ і знайти стягнення A в $G1$ ($G1 \setminus A$),
- добуток графів

20



2. Знайти таблицю суміжності та діаметр графа.

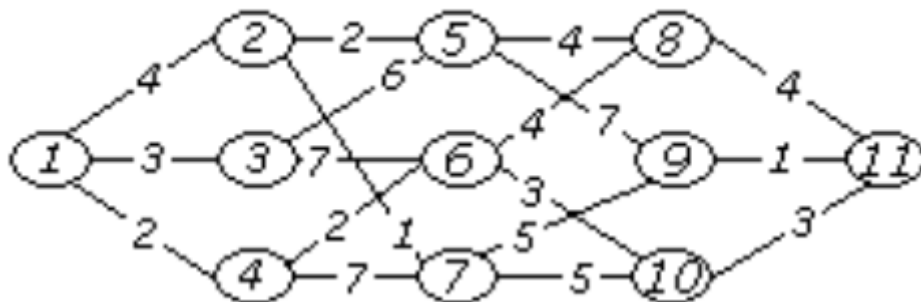
20



3. Для наступного графа реалізувати 3 алгоритми:

1. Знайти двома методами: Крaskала і Прима, мінімальне остовне дерево графа. По крокам розписати вибір ребер за кожним алгоритмом. Знайти вагу остовного дерева.
2. Знайти найкоротший ланцюг між вершинами 1 та 11 методом Дейкстра. Розписати алгоритм по крокам.

20

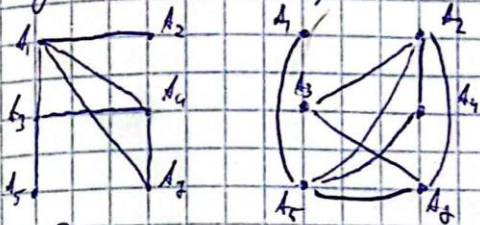


Розв'язок завдання №1

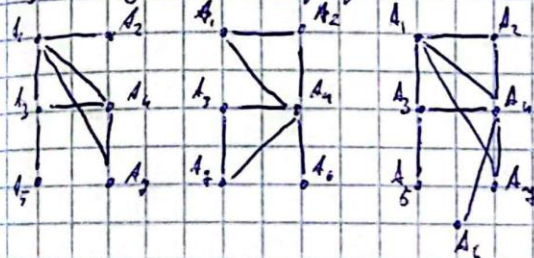
№1

ком
до 3

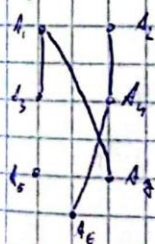
1) доповнення першого



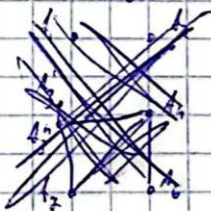
2) відзначити гради



3) числова сума



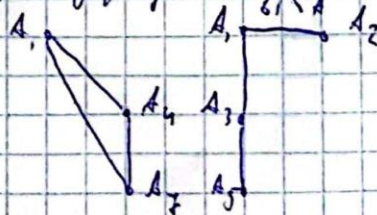
4) розрешена вершина



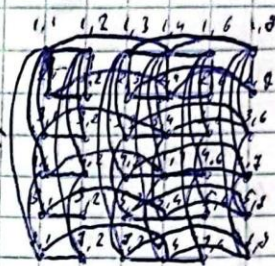
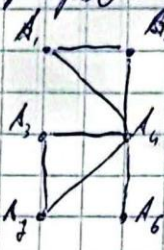
4) розрешена вершина A_3



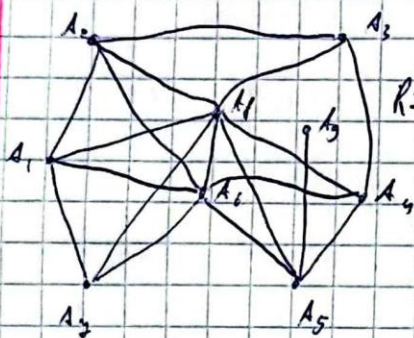
5) вершина A_1 , сума A_6 61



6) графы



✓ 2



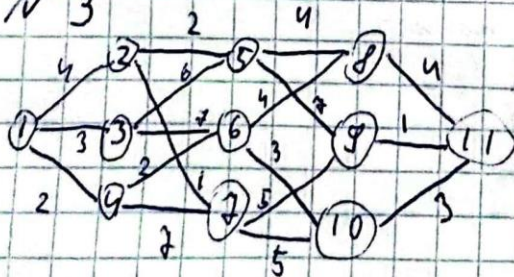
$$R = \begin{array}{c|cccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 4 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 5 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 6 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 7 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 8 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 9 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}$$

2

$$\begin{array}{c|cccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 1 & 1 & 0 & 1 & 2 & 2 & 2 & 1 & 1 & 3 \\ 2 & 0 & 1 & 2 & 3 & 1 & 2 & 1 & 4 \\ 3 & 0 & 1 & 2 & 2 & 2 & 1 & 3 \\ 4 & 0 & 1 & 1 & 2 & 1 & 2 \\ 5 & 0 & 1 & 2 & 2 & 4 \\ 6 & 0 & 1 & 1 & 2 \\ 7 & 0 & 1 & 3 \\ 8 & 0 & 3 \\ 9 & 0 \end{array}$$

$$d(6) = \max d(x_i, x_j) = 4$$

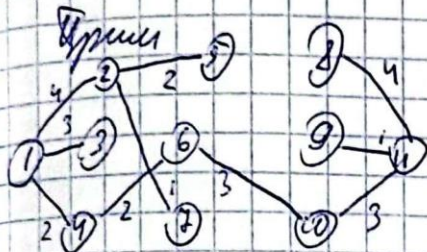
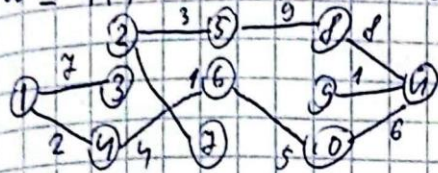
✓ 3



$$\begin{array}{l} w(1,3) = 3 \quad w(3,6) = 3 \\ w(2,7) = 1 \quad w(4,11) = 4 \quad w(4,7) = 1 \\ w(9,11) = 1 \quad w(5,8) = 4 \\ w(1,4) = 2 \quad w(6,8) = 4 \\ w(2,5) = 2 \quad w(2,9) = 5 \\ w(4,6) = 2 \quad w(1,10) = 5 \\ w(6,10) = 3 \quad w(3,5) = 6 \\ w(10,11) = 3 \quad w(5,9) = 2 \end{array}$$

$$n = |V| = 11$$

$$W = 1+1+2+2+2+3+3+3+4+4 = 25$$



1) Проверка формулы 1

$$\left. \begin{array}{l} 1-2=4 \\ 1-3=3 \\ 1-4=2 \end{array} \right\} = 1-4=2$$

$$\left. \begin{array}{l} 10) 9-5=2 \\ 1-5=4 \\ 2-5=2 \\ 3-5=6 \end{array} \right\} = 2-5=2$$

$$\left. \begin{array}{l} 3) 1-2=4 \\ 1-3=3 \\ 4-5=2 \\ 6-5=2 \\ 6-8=4 \\ 6-10=3 \end{array} \right\} = 6-10=3$$

$$\left. \begin{array}{l} 2) 1-2=4 \\ 1-3=3 \\ 4-6=2 \\ 4-8=2 \end{array} \right\} = 4-6=2$$

$$W = 2+2+3+3+4+1+4+1+3+2 = 25$$

$$\left. \begin{array}{l} 5) 11-8=4 \\ 11-9=1 \\ 6-8=4 \\ 6-3=1 \\ 1-2=4 \\ 1-3=3 \end{array} \right\} = 11-9=1$$

$$\left. \begin{array}{l} 4) 10-11=3 \\ 10-8=5 \\ 6-8=4 \\ 6-3=1 \\ 1-3=3 \\ 1-2=4 \end{array} \right\} = 10-11=3$$

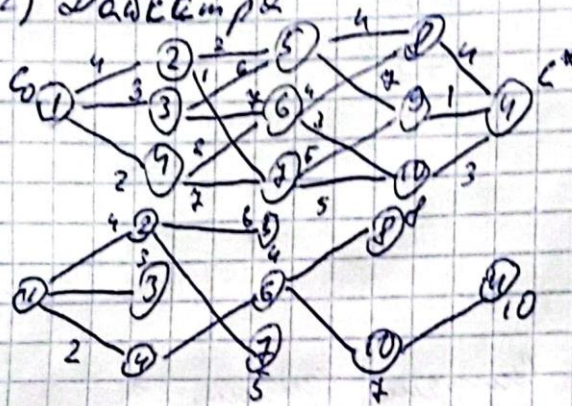
$$\left. \begin{array}{l} 7) 3-5=6 \\ 1-2=4 \\ 6-8=4 \\ 11-8=4 \\ 10-7=5 \\ 4-8=1 \end{array} \right\} = 11-8=4$$

$$\left. \begin{array}{l} 6) 11-8=4 \\ 9-5=2 \\ 9-7=2 \\ 8-8=4 \\ 6-3=1 \\ 1-2=4 \\ 1-3=3 \end{array} \right\} = 1-3=3$$

$$\left. \begin{array}{l} 9) 2-5=2 \\ 2-3=1 \\ 1-7=4 \\ 3-5=6 \\ 3-8=5 \end{array} \right\} = 2-3=1$$

$$\left. \begin{array}{l} 8) 8-5=4 \\ 1-2=4 \\ 3-5=6 \\ 1-3=3 \\ 3-5=6 \end{array} \right\} = 1-2=4$$

2) Dantzig



$$1) \left. \begin{array}{l} 1-2=4 \\ 1-3=3 \\ 1-7=2 \end{array} \right\} = 1-4=2$$

$$2) \left. \begin{array}{l} 4-6=4 \\ 4-8=5 \\ 1-2=1 \\ 1-3=3 \end{array} \right\} = 1-3=3$$

$$3) \left. \begin{array}{l} 3-6=10 \\ 3-5=9 \\ 4-6=4 \\ 1-8=9 \\ 1-2=1 \end{array} \right\} = 4-6=4$$

$$5) \left. \begin{array}{l} 2-8=5 \\ 3-5=9 \\ 6-8=8 \\ 6-10=7 \end{array} \right\} = 2-8=5$$

$$4) \left. \begin{array}{l} 1-2=4 \\ 3-5=9 \\ 6-8=8 \\ 6-10=7 \end{array} \right\} = 1-2=4$$

$$6) \left. \begin{array}{l} 2-5=6 \\ 4-8=10 \\ 7-10=10 \\ 6-10=3 \\ 6-8=10 \end{array} \right\} = 2-5=6$$

$$8) \left. \begin{array}{l} 5-8=10 \\ 5-9=13 \\ 2-9=10 \\ 6-8=8 \\ 10-4=10 \end{array} \right\} = 6-8$$

$$7) \left. \begin{array}{l} 5-8=10 \\ 5-9=13 \\ 6-8=8 \\ 6-10=7 \\ 7-9=10 \\ 7-10=10 \end{array} \right\} = 6-10=7$$

$$9) \left. \begin{array}{l} 1-11=12 \\ 10-11=10 \\ 7-9=10 \end{array} \right\} = 10-11=10$$

$1 \rightarrow 4 \rightarrow 6 \rightarrow 10 \rightarrow 11$ - найкоротший шлях

$$L = 2 + 2 + 3 + 3 = 10$$

Текст завдання №2

Написати програму, яка реалізує один з вивчених алгоритмів завдання 1.3: знаходження остовного дерева мінімальної ваги за алгоритмом Краскала, алгоритм Дейкстра пошуку найкоротшого ланцюга. Етапи розв'язання задачі виводити на екран. Протестувати розроблену програму на задачі 3 із завдання №1.

Розв'язок завдання №2

```
import networkx as nx
graph = nx.Graph()

num_vertices = int(input("\nВведіть кількість вершин: "))
print()

for i in range(1, num_vertices+1):
    for j in range(i+1, num_vertices+1):
        if i != j:
            weight = int(
                input(f"Введіть вагу між вершиною {i} та {j}: "))
            if weight != 0:
                graph.add_edge(i, j, weight=weight)

T = nx.minimum_spanning_tree(graph)
print(f"\nМінімальне остовне дерево:")

total_weight = 0
for u, v, data in T.edges(data=True):
    weight = data['weight']
    print(f"{u} -- {v} = {weight}")
    total_weight += weight
print(f"Вага дерева = {total_weight}")

def dijkstra(graph, start, end):
    dist = {v: float('inf') for v in graph.nodes()}
    dist[start] = 0
    prev = {}
    unvisited = set(graph.nodes())
    while unvisited:
        current = min(unvisited, key=lambda v: dist[v])
```

```

        unvisited.remove(current)
        if dist[current] == float('inf'):
            break
        for neighbor in graph.neighbors(current):
            tentative_dist = dist[current] +
graph[current][neighbor]['weight']
            if tentative_dist < dist[neighbor]:
                dist[neighbor] = tentative_dist
                prev[neighbor] = current

    path = []
    current = end
    while current != start:
        path.append(current)
        current = prev[current]
    path.append(start)
    path.reverse()
    path_str = " -> ".join(str(v) for v in path)
    return path_str

start = list(graph.nodes())[0]
end = list(graph.nodes())[-1]
shortest_path = dijkstra(graph, start, end)
print(f"\nНайкоротший шлях: {shortest_path}\n")

```

Результати роботи завдання №2

Введіть кількість вершин: 11

Введіть вагу між вершиною 1 та 2: 4
Введіть вагу між вершиною 1 та 3: 3
Введіть вагу між вершиною 1 та 4: 2
Введіть вагу між вершиною 1 та 5: 0
Введіть вагу між вершиною 1 та 6: 0
Введіть вагу між вершиною 1 та 7: 0
Введіть вагу між вершиною 1 та 8: 0
Введіть вагу між вершиною 1 та 9: 0
Введіть вагу між вершиною 1 та 10: 0
Введіть вагу між вершиною 1 та 11: 0
Введіть вагу між вершиною 2 та 3: 0
Введіть вагу між вершиною 2 та 4: 0
Введіть вагу між вершиною 2 та 5: 2
Введіть вагу між вершиною 2 та 6: 0
Введіть вагу між вершиною 2 та 7: 1
Введіть вагу між вершиною 2 та 8: 0
Введіть вагу між вершиною 2 та 9: 0
Введіть вагу між вершиною 2 та 10: 0
Введіть вагу між вершиною 2 та 11: 0
Введіть вагу між вершиною 3 та 4: 0
Введіть вагу між вершиною 3 та 5: 6
Введіть вагу між вершиною 3 та 6: 7
Введіть вагу між вершиною 3 та 7: 0
Введіть вагу між вершиною 3 та 8: 0
Введіть вагу між вершиною 3 та 9: 0
Введіть вагу між вершиною 3 та 10: 0
Введіть вагу між вершиною 3 та 11: 0
Введіть вагу між вершиною 4 та 5: 0
Введіть вагу між вершиною 4 та 6: 2
Введіть вагу між вершиною 4 та 7: 7
Введіть вагу між вершиною 4 та 8: 0
Введіть вагу між вершиною 4 та 9: 0
Введіть вагу між вершиною 4 та 10: 0
Введіть вагу між вершиною 4 та 11: 0
Введіть вагу між вершиною 5 та 6: 0
Введіть вагу між вершиною 5 та 7: 0
Введіть вагу між вершиною 5 та 8: 4
Введіть вагу між вершиною 5 та 9: 7
Введіть вагу між вершиною 5 та 10: 0
Введіть вагу між вершиною 5 та 11: 0
Введіть вагу між вершиною 6 та 7: 0
Введіть вагу між вершиною 6 та 8: 4
Введіть вагу між вершиною 6 та 9: 0
Введіть вагу між вершиною 6 та 10: 3
Введіть вагу між вершиною 6 та 11: 0
Введіть вагу між вершиною 7 та 8: 0
Введіть вагу між вершиною 7 та 9: 5
Введіть вагу між вершиною 7 та 10: 5
Введіть вагу між вершиною 7 та 11: 0
Введіть вагу між вершиною 8 та 9: 0
Введіть вагу між вершиною 8 та 10: 0
Введіть вагу між вершиною 8 та 11: 4
Введіть вагу між вершиною 9 та 10: 0
Введіть вагу між вершиною 9 та 11: 1
Введіть вагу між вершиною 10 та 11: 3

Мінімальне остовне дерево:

1 -- 4 = 2
1 -- 3 = 3
1 -- 2 = 4
2 -- 7 = 1
2 -- 5 = 2
4 -- 6 = 2
5 -- 8 = 4
6 -- 10 = 3
9 -- 11 = 1
10 -- 11 = 3

Вага дерева = 25

Найкоротший шлях: 1 -> 4 -> 6 -> 10 -> 11

Блок-схема завдання №2

