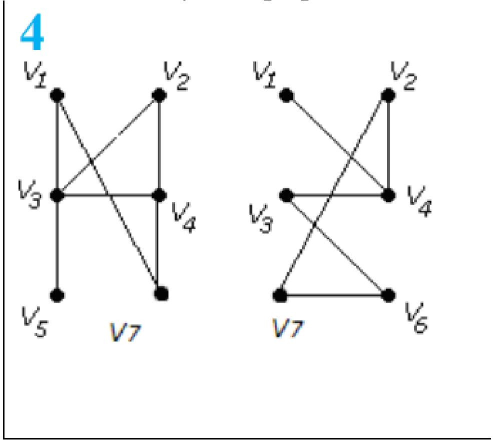
Лабораторна робота №5
Основні поняття теорії графів

Мета: забезпечити засвоєння студентами поняття графа, його основних елементів, навчитися визначати різні типи графів, знаходити степені вершин графа, застосовувати теорію графів для розв'язку прикладних задач.

Зміст роботи

Завдання 1. Розв'язати на графах наступні задачі:

1. знайти доповнення до першого графу;
2. знайти об'єднання графів;
3. кільцеву суму G1 та G2 (G1+G2);
4. розщепити вершину у другому графі;
5. виділити підграф A, що складається з 3-х вершин в G1 і знайти стягнення A в G1 (G1\ A);
6. знайти добуток графів.



Нехай дані два неорієнтовані прості графи G1 (лівий) і G2 (правий).

Зчитаємо графи з рисунка

G1:
V1 = {v1, v2, v3, v4, v5, v7}
E1 = { v1v3, v3v5, v3v4, v2v4, v4v7, v2v3, v1v7 }

G2:
V2 = {v1, v2, v3, v4, v6, v7}
E2 = { v2v4, v3v4, v1v4, v7v2, v3v6, v7v6 }
Доповнення до першого графу (комплемент) $\bar{G}1$

Для G1 маємо 6 вершин, отже всього можливих ребер: C(6,2) = 15.
У G1 є 7 ребер, тому в доповненні буде 15 − 7 = 8 ребер.

E($\bar{G}1$) = { v1v2, v1v4, v1v5, v2v5, v2v7, v3v7, v4v5, v5v7 }

Відповідь:
$\bar{G}1 = (V1, E(\bar{G}1))$, де V1 = {v1,v2,v3,v4,v5,v7}.

Об’єднання графів G = G1 ∪ G2

V(G) = V1 ∪ V2 = {v1, v2, v3, v4, v5, v6, v7}

E(G) = E1 ∪ E2
Спільні ребра (є в обох): v2v4, v3v4.

Отже:
E(G) = { v1v3, v3v5, v3v4, v2v4, v4v7, v2v3, v1v7, v1v4, v7v2, v3v6, v7v6 }

Кільцева сума (симетрична різниця) G1 + G2

G1 + G2 = (V1 ∪ V2, (E1 \ E2) ∪ (E2 \ E1))
Тобто беремо всі ребра, які належать рівно одному графу.

Оскільки спільні ребра: v2v4, v3v4, то їх прибираємо з об’єднання.

V = {v1, v2, v3, v4, v5, v6, v7}

E(G1+G2) = { v1v3, v3v5, v4v7, v2v3, v1v7, v1v4, v7v2, v3v6, v7v6 }

Розщепити вершину у другому графі (покажу один коректний варіант)

У G2 найбільш “центральна” вершина — v4 (вона має ребра з v1, v2, v3).
Розщепимо v4 на дві вершини v4a і v4b, розподіливши інцидентні ребра, і з’єднаємо їх між собою.

Нехай:

до v4a під’єднаємо ребра v1v4 і v3v4;

до v4b під’єднаємо ребро v2v4;

додамо ребро v4a v4b.

Тоді новий граф G2’:

V(G2’) = {v1, v2, v3, v4a, v4b, v6, v7}
E(G2’) = { v1v4a, v3v4a, v2v4b, v4a v4b, v7v2, v3v6, v7v6 }

Примітка: розщеплення не є єдиним — можна по-іншому розподілити ребра між v4a і v4b, але принцип той самий.

Виділити підграф A з 3 вершин у G1 і виконати стягнення A в G1 (G1 / A)

Візьмемо підграф A на вершинах {v1, v3, v5} (у G1 це зв’язний “ланцюжок” v1–v3–v5).

Стягнення: зливаємо вершини v1, v3, v5 в одну вершину a*.
Петлі прибираємо, кратні ребра зводимо до одного.

Зовнішні вершини (поза A): {v2, v4, v7}

Ребра між A і зовнішніми вершинами в G1:

v3v2 ⇒ a* v2

v3v4 ⇒ a* v4

v1v7 ⇒ a* v7

Ребра між зовнішніми вершинами, які були в G1:

v2v4 є

v4v7 є

Отже після стягнення:

V(G1/A) = {a*, v2, v4, v7}
E(G1/A) = { a* v2, a* v4, a* v7, v2v4, v4v7 }

Добуток графів

У вашому курсі під "добутком графів" найчастіше мають на увазі декартів (картизіанський) добуток G1 □ G2.

Визначення (картизіанський добуток):
V(G1 □ G2) = V1 × V2 (усі пари (u, v))
Ребро між (u, v) і (u', v') існує тоді і тільки тоді, коли:

або u = u' і vv' ∈ E2,

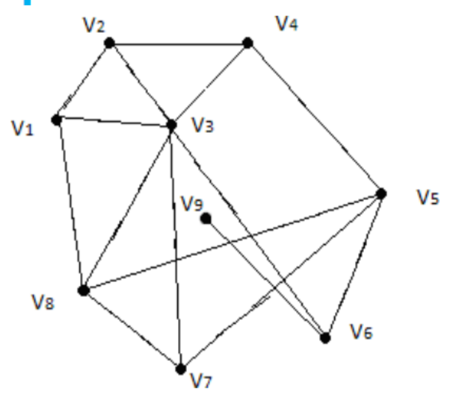
або v = v' і uu' ∈ E1.

Розміри:
|V1| = 6, |V2| = 6 ⇒ |V(G1 □ G2)| = 36

Кількість ребер у картизіанському добутку:
|E| = |E1|·|V2| + |E2|·|V1| = 7·6 + 6·6 = 42 + 36 = 78

Завдання 2. Знайти таблицю суміжності та діаметр графа.

4



Таблиця суміжності та діаметр графа

Приймемо порядок вершин:
v1, v2, v3, v4, v5, v6, v7, v8, v9

	v1	v2	v3	v4	v5	v6	v7	v8	v9
V1	0	1	1	0	0	0	0	1	0
V2	1	0	1	1	0	0	0	0	0
V3	1	1	0	1	0	1	1	1	0
V4	0	1	1	0	1	0	0	0	0
V5	0	0	0	1	0	1	1	1	0
V6	0	0	1	0	1	0	1	0	1
V7	0	0	1	0	1	1	0	1	0
V8	1	0	1	0	1	0	1	0	0
V9	0	0	0	0	0	1	0	0	0

Діаметр графа

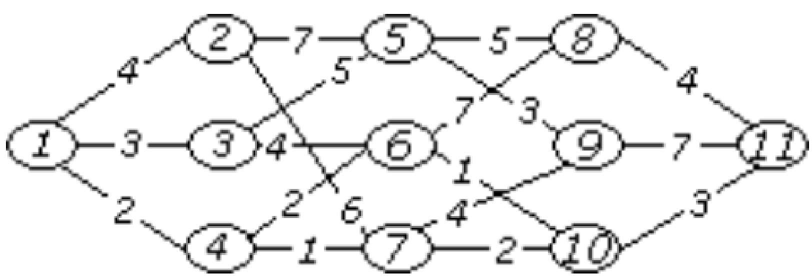
Діаметр — це найбільша довжина найкоротшого шляху між будь-якими двома вершинами графа.

Найбільша відстань у цьому графі дорівнює 3 (наприклад, між v1 і v9: v1–v3–v6–v9).

Відповідь:
D(G) = 3

Завдання 3. Знайти двома методами (Краскала і Прима) мінімальне остоведерево графа.

4



Завдання 4. Написати програму, яка реалізує алгоритм знаходження остового дерева мінімальної ваги за алгоритмом Прима чи Краскала. Етапи розв'язання задачі виводити на екран. Протестувати розроблену програму на задачі 3 із завдання № 1.

https://github.com/ShadowGhost31/DSKMLabs/tree/main/Lab5_Task4