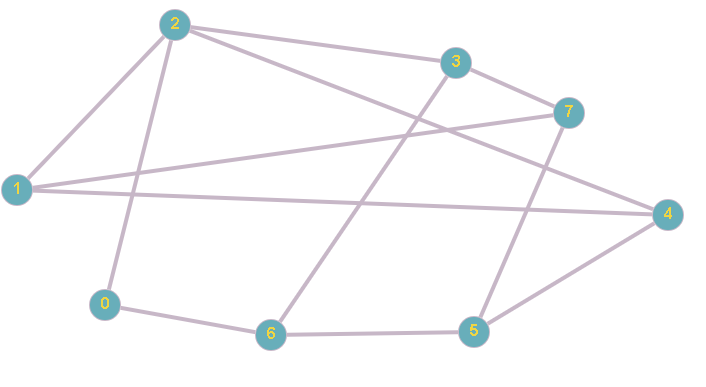
**Лабораторна робота 3**

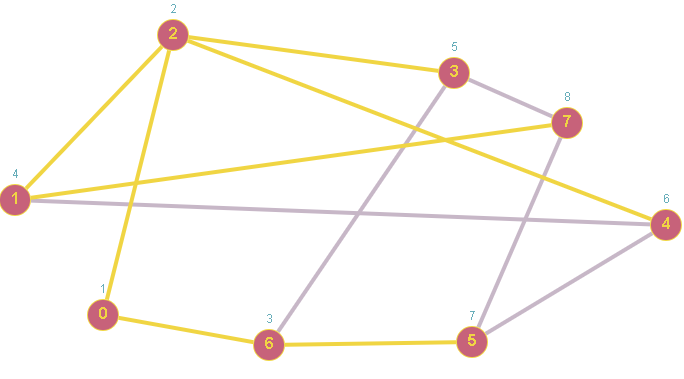
**Тема: Методи обходу та модифікації графів**

**Мета роботи: Навчитись застосовувати алгоритми обходу графів, побудови дерева шляхів та мінімального зв’язного дерева**

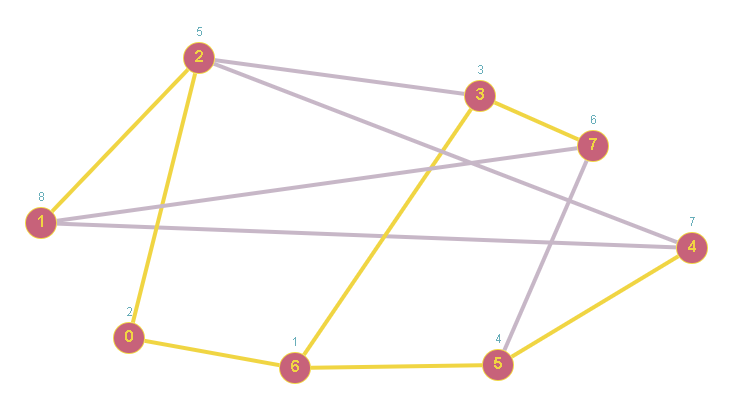
**Хід роботи**

1.Побудовано неорієнтований граф G = {8,12}: [](https://user-images.githubusercontent.com/79449241/118645221-536e5480-b7e7-11eb-9f55-1a8f0c75c9ca.png)

i.Побудувано дерево за алгоритмом обходу в ширину (BFS) для вершини 0:

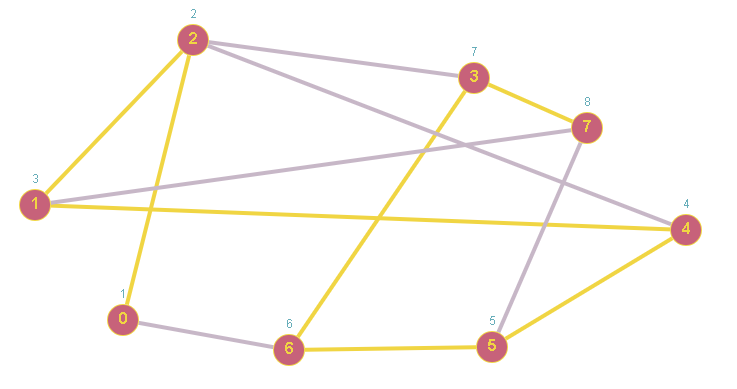
[](https://user-images.githubusercontent.com/79449241/118645729-f58e3c80-b7e7-11eb-90df-1aee782dc06d.png)

для вершини 6:

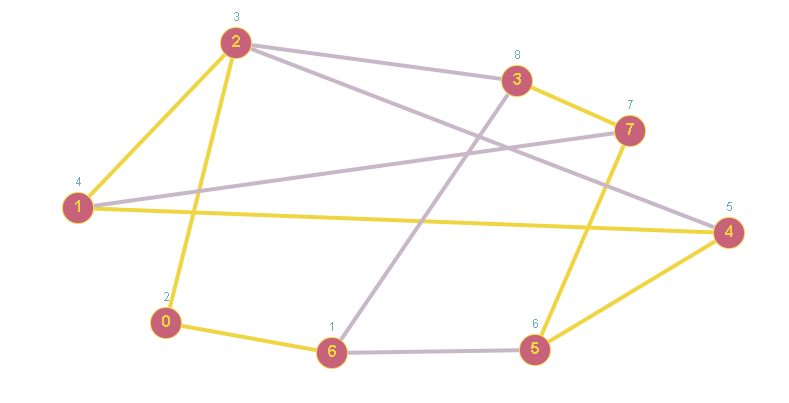
[](https://user-images.githubusercontent.com/79449241/118645657-dee7e580-b7e7-11eb-9897-7f890626b8b1.png)

ii. Топології дерев побудованих з різних кореневих вершин будуть різними. Це пояснюється тим, що алгоритм обходить не всі ребра графу, відповідно якщо у графі є вершини які мають більше трьох суміжних вершин, то тополігї дерев побудованих з різних кореневих вершин будуть різними.

iii. Побудувано дерево за алгоритмом обходу в глибину (DFS); з вершини 0:

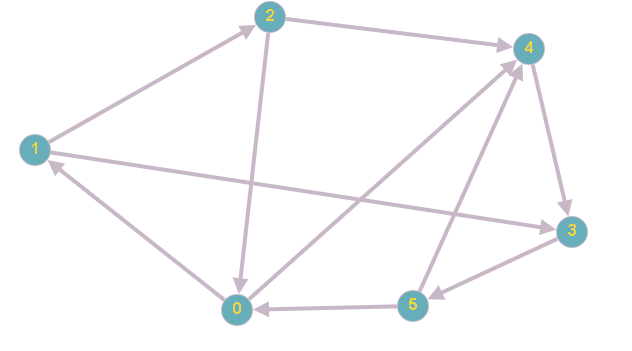
[](https://user-images.githubusercontent.com/79449241/118646931-769a0380-b7e9-11eb-97ea-8bac0e30d8a6.png)

з вершини 6:

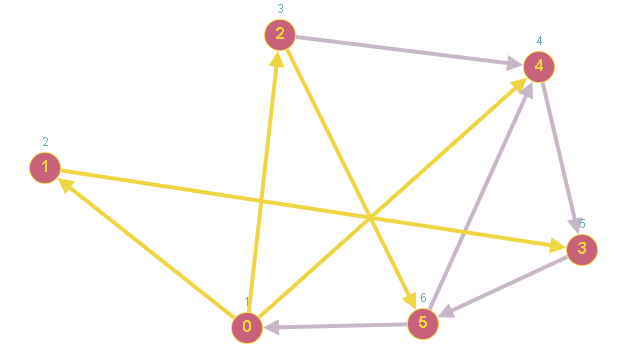
[](https://user-images.githubusercontent.com/79449241/118647070-99c4b300-b7e9-11eb-8367-9bdbd6510712.png)

iv. Топології будуть різними. Це можна пояснити тим, що на кожному кроці алгоритм випадково знаходить лише одну вершину( випадково вибираючи одну з тих, що до неї під'єднані).

1. Побудувано випадковий орієнтований граф G={6,10}:

[](https://user-images.githubusercontent.com/79449241/118647584-2bccbb80-b7ea-11eb-87f1-dd22e2ff39f7.png)

i. Побудувано дерево за алгоритмом обходу в ширину (BFS), починаючи з нульової вершини:

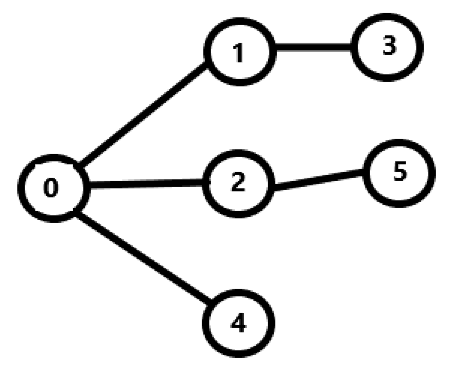
[](https://user-images.githubusercontent.com/79449241/118648096-bca39700-b7ea-11eb-952e-529e19ed20eb.png)

Порядок обходу: 0 1 2 4 3 5

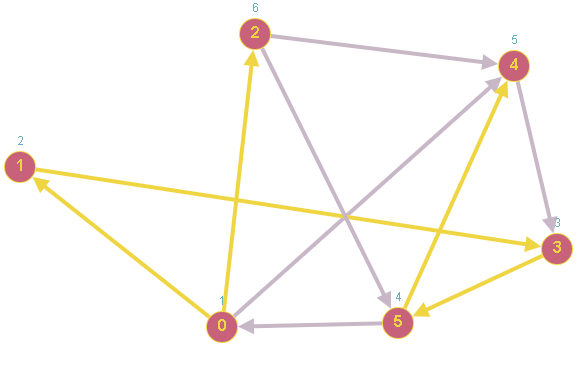
ii. Якщо починати обхід графу за алгоритмом обходу в ширину з нульової вершини, то останньою буде знайдена вершина п'ята.

iii. Граф не містить циклів Ейлера та Гамільтона.

iv. Для виявлення останньої вершини потрібно три хвилі

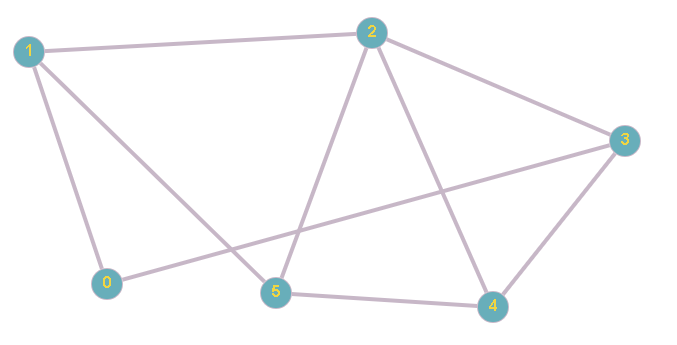
[](https://user-images.githubusercontent.com/79449241/118649101-ca0d5100-b7eb-11eb-9b08-0a4a15253ca1.png)

v. Дерево за алгоритмом обходу в глибину (DFS):

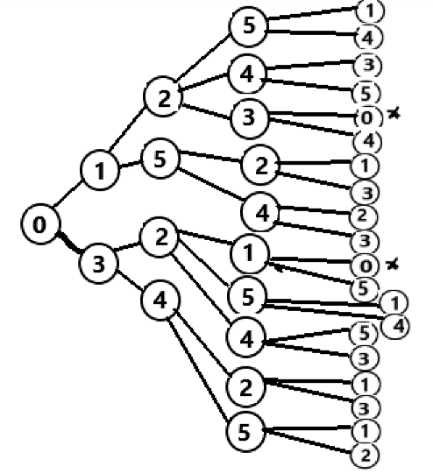
[](https://user-images.githubusercontent.com/79449241/118649499-3daf5e00-b7ec-11eb-9fc0-2a92105df4b4.png)

Порядок обходу: 0 1 3 5 4 2

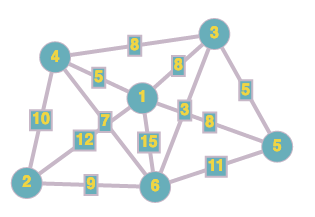
1. Побудувати дерево шляхів рангом r=4 для випадкового графа G={6,9}.

[](https://user-images.githubusercontent.com/79449241/118651508-50c32d80-b7ee-11eb-90a8-e639fe368ffe.png)

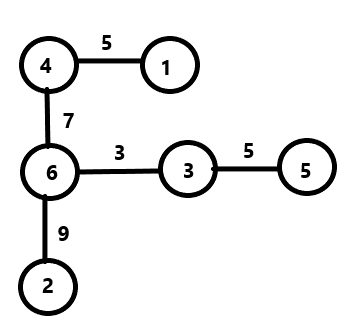
дерево шляхів:

[](https://user-images.githubusercontent.com/79449241/118657184-ec0ad180-b7f3-11eb-9767-159bd8a6d274.png)

1. Побудувати мінімальне зв’язне дерево для графа G. Вказати його вагу.

[](https://user-images.githubusercontent.com/79449241/118650686-6421c900-b7ed-11eb-9d12-579fb33b5333.png)

Мінімальне зв'язне дерево:

[](https://user-images.githubusercontent.com/79449241/118657228-f75dfd00-b7f3-11eb-9a69-e599cada7803.png)

Вага мінімального зв'язного дерева = 29.