

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



**Кафедра: САП**

Звіт до виконаної лабораторної роботи №4

з дисципліни “Дискретна математика”

на тему: "Ейлерові та гамільтонові графи. Алгоритм Дейкстри"

*Виконав:*

*студент групи ПП-16*

*Якіб'юк Ігор*

*Прийняв:*

*Асистент каф. САП*

*Іванина В. В.*

*Львів - 2023*

## Лабораторна робота № 4

**Мета роботи:** Мета роботи – вивчити можливості пакету GraphTheory системи Maple для відшукування ейлерових шляхів і циклів та знаходження найкоротшого шляху в графі за допомогою алгоритму Дейкстри.

### Хід роботи: Варіант 29

#### Лабіринт:

##### *Тренувальне завдання.*

Намалюйте граф, який відповідає лабіринту на рис. 11. Знайдіть шлях, за яким можна пройти від пункту А до В лабіринту, використовуючи запропонований вище алгоритм.

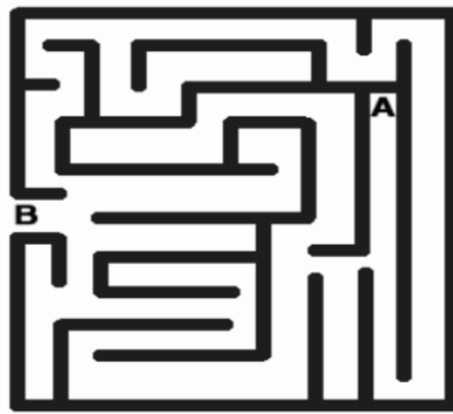
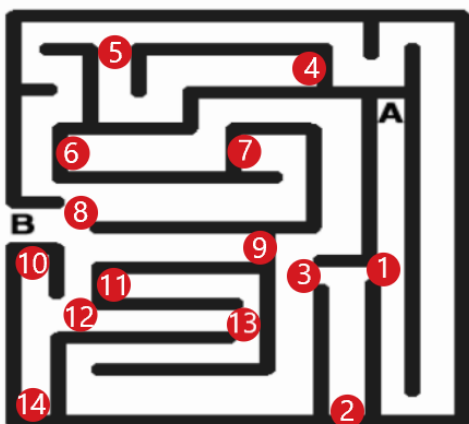


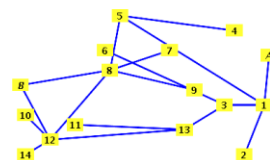
Рис. 11. Схема лабіринту від пункту А до В.

#### Розв'язання:



```
> M := AddEdge(M, {{A, 1}, {1, 2}, {1, 3}, {3, 6}, {3, 13}, {1, 5}, {5, 4}, {5, 8}, {8, B},  
{8, 7}, {8, 9}, {8, 12}, {12, 10}, {12, 14}, {12, 13}, {13, 11}, {12, B}});  
M := Graph 275: an undirected unweighted graph with 16 vertices and 17 edge(s)
```

```
> SetVertexPositions(M, vpAB);  
> DrawGraph(M);
```



```
> Distance(M, A, B);
```

4

Рис. 11. Схема лабіринту від пункту А до В. > ShortestPath(M, A, B);

[A, 1, 5, 8, B]

**Завдання 4.1.** Перевірити чи заданий граф є ейлеровим.

**Завдання 4.2.** Перевірити чи заданий граф є гамільтоновим.

**Завдання 4.3.** Побудувати ейлеровий та гамільтоновий графи (не менше 5 вершин).

**Завдання 4.4.** Знайти найкоротший шлях за допомогою алгоритму Дейкстри в неорієнтованому графі між двома вершинами (вибрати самостійно).

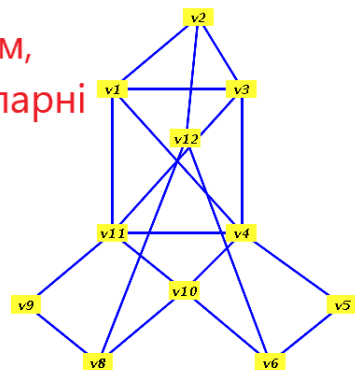
**Завдання 4.5.** Знайти найкоротший шлях в неорієнтованому графі від вершини 5 до всіх інших вершин.

**Завдання 4.6.** Знати найкоротший шлях у зваженому графі між трьома парами вершин (вибрати самостійно).

#### Розв'язання 4.1:

► *DrawGraph(G);*

Граф не є Ейлеровим,  
тому що містить непарні  
степені вершин



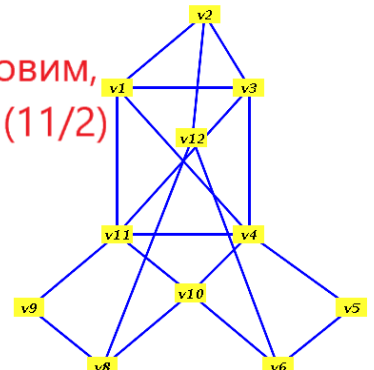
► *DegreeSequence(G);*

[4, 3, 4, 5, 2, 3, 3, 2, 4, 5, 3]

#### Розв'язання 4.2:

► *DrawGraph(G);*

Граф не є гамільтоновим,  
оскільки  $\text{degree}(v) < (11/2)$



► *DegreeSequence(G);*

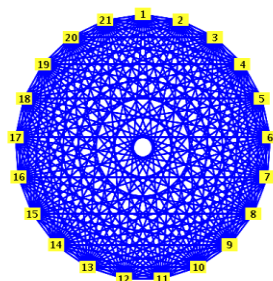
[4, 3, 4, 5, 2, 3, 3, 2, 4, 5, 3]

#### Розв'язання 4.3:

► *K := CompleteGraph(21);*

*K := Graph 272: an undirected unweighted graph with 21 vertices and 210 edge(s)*

► *DrawGraph(K);*



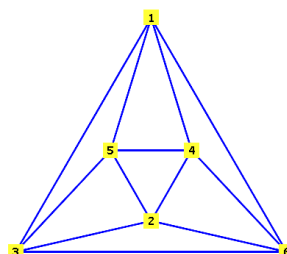
► *DegreeSequence(K);*

[20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20]

*J := OctahedronGraph( );*

*J := Graph 273: an undirected unweighted graph with 6 vertices and 12 edge(s)*

*DrawGraph(J);*



*DegreeSequence(J);*

[4, 4, 4, 4, 4, 4]

## Розв'язання 4.4 і 4.5:

*DijkstrasAlgorithm*(*G*, *v2*, *v6*);

[[*v2*, *v12*, *v6*], 2]

*DijkstrasAlgorithm*(*G*, *v5*, *v1*);

[[*v5*, *v4*, *v1*], 2]

*DijkstrasAlgorithm*(*G*, *v5*, *v2*);

[[*v5*, *v4*, *v3*, *v2*], 3]

*DijkstrasAlgorithm*(*G*, *v5*, *v3*);

[[*v5*, *v4*, *v3*], 2]

*DijkstrasAlgorithm*(*G*, *v5*, *v4*);

[[*v5*, *v4*], 1]

*DijkstrasAlgorithm*(*G*, *v5*, *v6*);

[[*v5*, *v6*], 1]

*DijkstrasAlgorithm*(*G*, *v5*, *v8*);

[[*v5*, *v6*, *v10*, *v8*], 3]

*DijkstrasAlgorithm*(*G*, *v5*, *v9*);

[[*v5*, *v4*, *v11*, *v9*], 3]

*DijkstrasAlgorithm*(*G*, *v5*, *v10*);

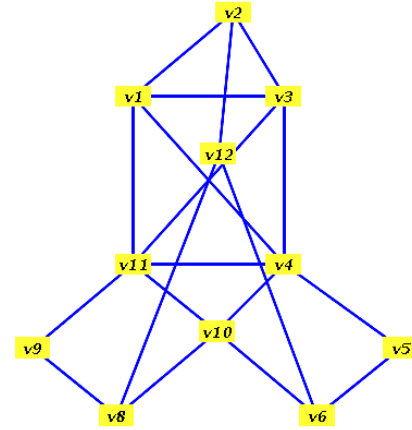
[[*v5*, *v6*, *v10*], 2]

*DijkstrasAlgorithm*(*G*, *v5*, *v11*);

[[*v5*, *v4*, *v11*], 2]

*DijkstrasAlgorithm*(*G*, *v5*, *v12*);

[[*v5*, *v6*, *v12*], 2]



## Розв'язання 4.6:

• *GGraph* := *Graph*([{ {1, 2}, 3}, { {2, 3}, 5}, { {1, 7}, 4}, { {2, 8}, 3}, { {3, 4}, 7}, { {4, 5}, 8}, { {5, 6}, 5}, { {7, 13}, 2}, { {7, 8}, 6}, { {3, 9}, 3}, { {4, 10}, 4}, { {5, 11}, 4}, { {6, 12}, 1}, { {11, 12}, 8}, { {10, 11}, 2}, { {9, 10}, 1}, { {8, 9}, 3}, { {8, 14}, 1}, { {13, 14}, 4}, { {9, 15}, 7}, { {14, 15}, 2}, { {10, 16}, 4}, { {15, 16}, 2}, { {16, 17}, 7}, { {17, 18}, 7}, { {19, 20}, 1}, { {20, 21}, 3}, { {21, 22}, 1}, { {22, 23}, 1}, { {23, 24}, 5}, { {24, 30}, 8}, { {25, 26}, 3}, { {26, 27}, 3}, { {27, 28}, 3}, { {28, 29}, 3}, { {29, 30}, 1}, { {12, 18}, 7}, { {18, 24}, 7}, { {13, 19}, 5}, { {19, 25}, 4}, { {14, 20}, 1}, { {20, 26}, 2}, { {15, 21}, 6}, { {16, 22}, 7}, { {17, 23}, 7}, { {18, 24}, 7}, { {23, 29}, 1}, { {20, 26}, 2}, { {21, 27}, 3}, { {22, 28}, 8}, { {11, 17}, 1}]);

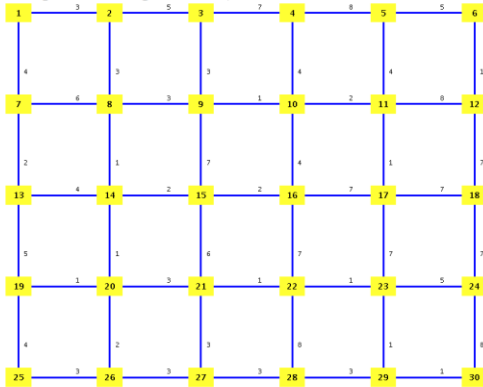
*GGraph* := *Graph* 274: an undirected weighted graph with 30 vertices and 49 edge(s) (

• *vp* := [[0, 4], [1, 4], [2, 4], [3, 4], [4, 4], [5, 4], [0, 3], [1, 3], [2, 3], [3, 3], [4, 3], [5, 3], [0, 2], [1, 2], [2, 2], [3, 2], [4, 2], [5, 2], [0, 1], [1, 1], [2, 1], [3, 1], [4, 1], [5, 1], [0, 0], [1, 0], [2, 0], [3, 0], [4, 0], [5, 0]];

*p* := [[0, 4], [1, 4], [2, 4], [3, 4], [4, 4], [5, 4], [0, 3], [1, 3], [2, 3], [3, 3], [4, 3], [5, 3], [0, 2], [1, 2], [2, 2], [3, 2], [4, 2], [5, 2], [0, 1], [1, 1], [2, 1], [3, 1], [4, 1], [5, 1], [0, 0], [1, 0], [2, 0], [3, 0], [4, 0], [5, 0]] (

• *SetVertexPositions*(*GGraph*, *vp*);

• *DrawGraph*(*GGraph*, *showweights* = true);



• *DijkstrasAlgorithm*(*GGraph*, 1, 30);

[[1, 2, 8, 14, 20, 21, 22, 23, 29, 30], 15] (

• *DijkstrasAlgorithm*(*GGraph*, 7, 24);

[[7, 13, 14, 20, 21, 22, 23, 24], 17] (

• *DijkstrasAlgorithm*(*GGraph*, 2, 28);

[[2, 8, 14, 20, 26, 27, 28], 13] (

**Висновок:** На даній лабораторній роботі я навчився шукати ейлерові шляхи, цикли та знаходити найкоротший шлях в графі за допомогою алгоритму Дейкстри в середовищі Maple18. Також виконав завдання з лабіринтом, використовуючи запропонований інструкцією алгоритм. Дослідив чи граф є ейлеровим або ж гамільтоновим. В наступному завданні побудував власні графи, які містили в собі гамільтоновий і ейлеровий цикли. Останнім завданням, скориставшись алгоритмом Дейкстри, знайшов найкоротший шлях між двома вершинами у графі. Виконані завдання продемонстрував вище.