Затверджую
Завідувач кафедри
прикладної математики і
механіки
ЛДУ БЖД

"__" ____________20___ р.

МЕТОДИЧНА РОЗРОБКА

ДЛЯ ПРОВЕДЕННЯ ПРАКТИЧНОГО ЗАНЯТТЯ
З КУРСАНТАМИ ТА СТУДЕНТАМИ 2 *КУРСУ*
З ДИСЦИПЛІНИ СИСТЕМНИЙ АНАЛІЗ ТА ТЕОРІЯ ПРИЙНЯТТЯ РІШЕНЬ

ТЕМА: № 1. Основні властивості та зображення графів. Алгоритм Фльорі.

Методична розробка обговорена на засіданні кафедри
Протокол № ____ від _______20 р.

м. Львів

ТЕМА: № 1. Основні властивості та зображення графів. Алгоритм Фльорі

Мета заняття

навчальна: ознайомити студентів з основними властивостями та зображеннями графів; навчити знаходити Ейлерів цикл (шлях) або доведити, що він не існує; навчити використовувати алгоритм Фльорі для знаходження Ейлеревого циклу у графі.

виховна: виховання свідомого ставлення до вивчення предмету, самостійності, відповідальності та організованості при підготовці до занять.

розвиткова: розвиток логічного та абстрактного мислення, розвиток просторової уяви.

Навчальний час: 2 години.
Місце проведення: згідно з розкладом.
Забезпечення заняття: ПК, МП.

Література:

1. *Махней О.В., Гой Т.П.* (2013). Математичне забезпечення автоматизації прикладних досліджень. Івано-Франківськ: Сімик, 304.

Структурні елементи заняття

- організаційно-вступна частина;
- вивчення нового матеріалу з одночасним виконанням прикладів на ПК;
- закріплення нового матеріалу;
- видача завдання для самостійного виконання.

Розробила:
доцент кафедри прикладної математики і механіки,
к. ф.-м. наук Оксана Чмир

Основні властивості графів (неорієнтованих та орієнтованих).

Простим графом називають пару $G = (V, E)$, де V – непорожня скінченна множина елементів, названих *вершинами*, E – множина невпорядкованих пар різних елементів з V. Елементи множини E (невпорядковані пари різних вершин) називають *ребрами.*

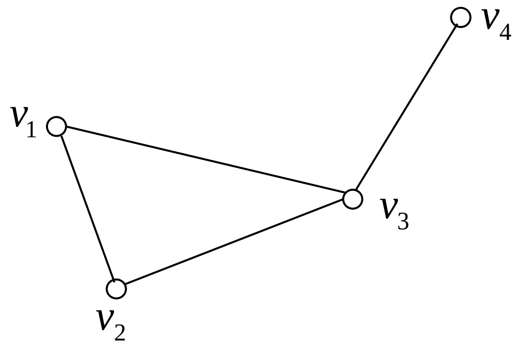


Рис. 1

Наприклад, на рис. 1 зображено простий граф G з множиною вершин $V = \{v_1, v_2, v_3, v_4\}$ і множиною ребер $E = \{\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_3\}, \{v_3, v_4\}\}$.

Говорять, що ребро $\{u, v\}$ *з'єднує* вершини u та v. Оскільки E – множина, то в простому графі пару вершин може з'єднувати не більше ніж одне ребро.

Розглядають також орієнтовані графи.

Орієнтованим графом називають пару (V, E), де V – скінченна непорожня множина вершин, а E – множина впорядкованих пар елементів множини V. Елементи множини E в орієнтованому графі називають *дугами* (*орієнтованими ребрами*).

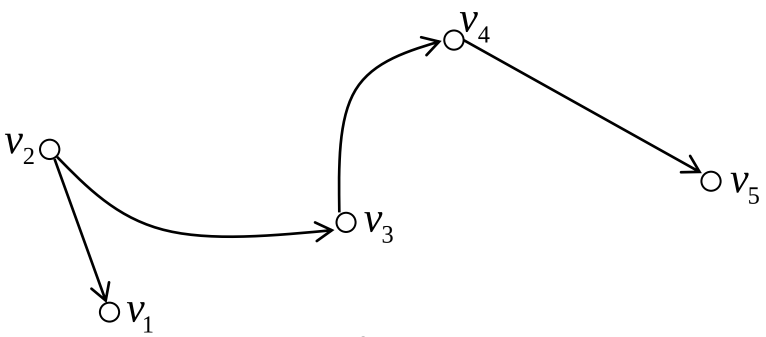


Рис. 2

Наприклад, на рис. 2 зображено орієнтований граф із множиною вершин $V = \{v_1, v_2, v_3, v_4, v_5\}$ і множиною дуг $E = \{(v_2, v_1), (v_2, v_3), (v_3, v_4), (v_4, v_5)\}$. На рисунках дуги позначають стрілками.

Дві вершини u та v в неорієнтованому графі G називають *суміжними*, якщо існує ребро $\{u, v\}$, тобто $\{u, v\} \in E$. Якщо $e = \{u, v\}$ – ребро, то вершини u та v називають його *кінцями.*

Два ребра називають *суміжними*, якщо вони мають спільний кінець. Вершину v та ребро e називають *інцидентними*, якщо вершина v – кінець ребра e.

Степінь вершини в неорієнтованому графі – це кількість ребер, інцидентних цій вершині. Степінь вершини v позначають $\deg(v)$. Якщо $\deg(v) = 0$, то вершину v називають *ізольованою*; якщо $\deg(v) = 1$ – *висячою* (*кінцевою*).

Наприклад, у неорієнтованому графі на рис. 3 степені вершин такі: $\deg(v_1) = 3$, $\deg(v_2) = 3$, $\deg(v_3) = 4$, $\deg(v_4) = 1$, $\deg(v_5) = 3$, $\deg(v_6) = 0$. Отже, вершина v_6 – ізольована, а v_4 – висяча.

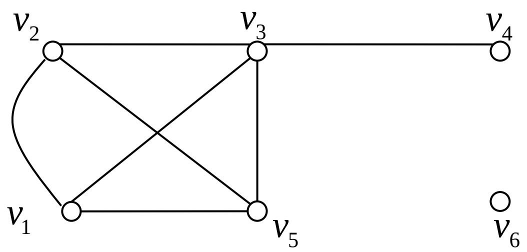


Рис. 3

Зв'язок між степенями вершин неорієнтованого графа та кількістю його ребер дає така теорема.

Теорема 1. Нехай $G = (V, E)$ – неорієнтований граф з m ребрами. Тоді

$$\sum_{v \in V} \deg(v) = 2m.$$

Теорема 2. Неорієнтований граф має парну кількість вершин непарного степеня.

Тепер розглянемо орієнтований мультиграф $G = (V, E)$.

Якщо $(u, v) \in E$, то вершину u називають *початковою*, а вершину v – *кінцевою* вершиною дуги $e = (u, v)$.

Вершини орієнтованого графа називають *суміжними*, якщо одна з них – початкова, а інша – кінцева для якоїсь дуги.

Дуги називають *суміжними*, якщо вони мають спільну вершину.

Вершину u називають *інцидентною* дузі e, якщо u – початкова чи кінцева вершина цієї дуги.

Для орієнтованого графа означення степеня вершини інше.

В орієнтованому мультиграфі *напівстепенем входу* вершини v називають кількість дуг, для яких вершина v кінцева; позначають $\deg^-(v)$.

Напівстепенем виходу вершини v називають кількість дуг, для яких вершина v початкова; позначають $\deg^+(v)$.

Зображення графів матрицями інцидентності та суміжності.

Найзрозуміліший і корисний спосіб подання (зображення) графів – це рисунок на площині у вигляді точок і ліній, які з'єднують ці точки. Проте цей спосіб подання абсолютно непридатний, якщо потрібно розв'язувати на комп'ютері задачі з графами.

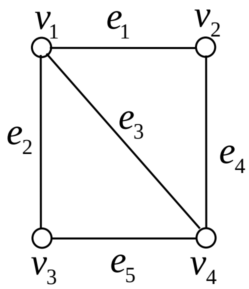


Рис. 4

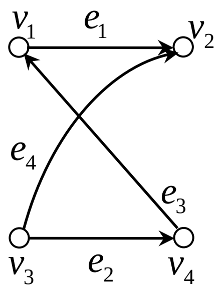


Рис. 5

Розглянемо декілька інших способів подання графів. Для спрощення розглядатимемо два найбільш важливих типи графів: простий граф (рис. 4) і орієнтований граф (рис. 5).

Матрицю, кожний елемент якої дорівнює 0 або 1, називають *булевою.*

Нехай $G = (V, E)$ – простий граф із множиною вершин $V = \{v_1, v_2, \ldots, v_n\}$ і множиною ребер $E = \{e_1, e_2, \ldots, e_m\}$.

Матрицею інцидентності графа G, яка відповідає заданій нумерації вершин і ребер, називають булеву $n \times m$ матрицю M з елементами m_{ij} ($i = \overline{1,n}$, $j = \overline{1,m}$), де

$$m_{ij} = \begin{cases} 1, & \textit{якщо вершина } v_i \textit{ та ребро } e_j \textit{ інцидентні}, \\ 0 & \textit{в протилежному випадку}. \end{cases}$$

Наприклад, для графа, зображеного на рис. 4, матриця інцидентності має вигляд

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array}.$$

Отже, для простого графа в матриці інцидентності в кожному стовпці точно дві одиниці, і немає однакових стовпців.

За допомогою матриці інцидентності можна подавати й орієнтовані графи. Для таких графів вона вже буде не булевою.

Нехай $G = (V, E)$ – орієнтований граф із множиною вершин $V = \{v_1, v_2, \ldots, v_n\}$ і множиною дуг $E = \{e_1, e_2, \ldots, e_m\}$.

Матрицею інциндентності орієнтованого графа G, яка відповідає заданій нумерації вершин і дуг, називають $n \times m$ матрицю M з елементами m_{ij} ($i = \overline{1, n}$, $j = \overline{1, m}$), де

$$m_{ij} = \begin{cases} 1, & \textit{якщо дуга } e_j \textit{ виходить з вершини } v_i, \\ -1, & \textit{якщо дуга } e_j \textit{ входить у вершину } v_i, \\ 0, & \textit{в інших випадках.} \end{cases}$$

Наприклад, для графа, зображеного на рис. 5, матриця інцидентності має вигляд

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & e_4 \end{array} \\ \begin{bmatrix} 1 & 0 & -1 & 0 \\ -1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \end{array}.$$

Нехай $G = (V, E)$ – простий граф, $|V| = n$. Припустимо, що вершини графа G занумеровані: $v_1, v_2, \dots, v_n$.

Матрицею суміжності графа G (яка відповідає даній нумерації вершин) називають булеву $n \times n$ матрицю A з елементами a_{ij} $(i, j = \overline{1, n})$, де

$$a_{ij} = \begin{cases} 1, & \textit{якщо } \{v_i, v_j\} \in E, \\ 0 & \textit{в протилежному випадку.} \end{cases}$$

Наприклад, матриця суміжності для графа, зображеного на рис. 4, має вигляд

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array}.$$

Цілком очевидно, що для неорієнтованого графа $a_{ij} = a_{ji}$, тобто матриця суміжності неорієнтованого графа симетрична. Більше того, позаяк у простому графі немає петель, то для нього в матриці суміжності $a_{ii} = 0$ ($i = \overline{1, n}$).

Для подання орієнтованих графів також використовують матрицю суміжності.

Матрицею суміжності орієнтованого графа G називають булеву $n \times n$ матрицю A з елементами a_{ij} $(i, j = \overline{1, n})$, де

$$a_{ij} = \begin{cases} 1, & \textit{якщо } (v_i, v_j) \in E, \\ 0 & \textit{в протилежному випадку.} \end{cases}$$

Наприклад, матриця суміжності для графа, зображеного на рис. 5, має вигляд

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{array}.$$

Зазначимо, що <u>матриця суміжності орієнтованого графа, загалом кажучи, несиметрична</u>.

Поняття Ейлерового циклу (шляху). Алгоритм Фльорі.

Ейлеровим циклом у графі G називають простий цикл, який містить усі ребра графа.

Ейлеровим шляхом у графі G називають простий шлях, який містить усі ребра графа.

Наприклад, на рис. 6 граф G_1 має ейлерів цикл: a, e, c, d, e, b, a; граф G_3 не має ейлерового циклу, але має ейлерів шлях: a, c, d, e, b, d, a, b; граф G_2 не має ні ейлерового циклу, ні ейлерового шляху.

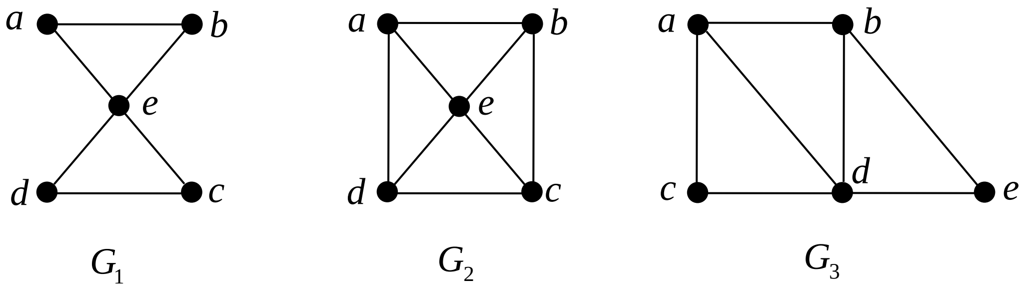


Рис. 6

Існує простий критерій (необхідна й достатня умова) для виявлення, чи має граф ейлерів цикл.

Теорема 3. Зв’язний граф має ейлерів цикл тоді й лише тоді, коли степені всіх його вершин парні.

Одним із алгоритмів знаходження Ейлеревого циклу у графі є <u>**алгоритм Фльорі**</u>, який ґрунтується на двох правилах:

1. Починаємо з довільної вершини, проходимо по ребрах графа довільним чином. При проходженні кожного ребра йому присвоюється черговий номер в циклі і це ребро викреслюється, тобто вилучається з графа (вершини при цьому лишаються);

2. Якщо є декілька можливостей продовжити рух по ребрах, то рухаємось не по ребру, при вилученні якого граф перестає бути зв’язним.

Теорема 4. Зв’язний граф має ейлерів шлях, але не має ейлерового циклу тоді й лише тоді, коли він має точно дві вершини непарного степеня.

Зазначимо, що *будь-який ейлерів шлях починається в одній із цих двох вершин непарного степеня, а закінчується в іншій.*

Граф, який має ейлерів цикл, часто називають *ейлеровим графом.*

Поняття Гамільтонового циклу (шляху).

Шлях x_0, x_1, …, x_{n-1} у неорієнтованому зв'язному графі $G=(V, E)$, $V=\{v_1,\ldots, v_n\}$ називають *гамільтоновим шляхом*, якщо $x_i \in V$ і $x_i \neq x_j$ для $0 \le i < j \le n-1$.

Цикл x_0, x_1, …, x_{n-1}, x_0 (тут $n \ge 3$) у графі G називають *гамільтоновим циклом*, якщо x_0, x_1, …, x_{n-1} – гамільтонів шлях.

Інакше кажучи, гамільтонів цикл і гамільтонів шлях проходять через кожну вершину графа точно один раз (у циклі окрім першої й останньої вершин). Зазначимо, що гамільтонові цикл і шлях, узагалі кажучи, не містять усіх ребер графа. За означенням гамільтонові цикли розглядають лише для графів, які мають не менше трьох вершин (отже, довжина гамільтонового циклу не менша трьох).

Розв'язання задач теорії графів в Maple

Для розв'язання задач теорії графів в ***Maple*** потрібно підключити пакет ***GraphTheory***:

>with(***GraphTheory***):

Пакет містить понад 150 функцій, детальніше з якими можна познайомитись в довідковій системі ***Maple***.

Зокрема, зважений граф (орієнтований та неорієнтований) можна задати функцією

***Graph*(*A*)**

де ***A*** – матриця довжин дуг (ребер) графа. Якщо дуга (ребро) (i, j) відсутня у графі, то елемент матриці $a_{ij}=0$.

Зобразити граф можна за допомогою функції

***DrawGraph*(*G*)**

де ***G*** – заданий граф (див. рис. 7).

```
> restart :
> with(GraphTheory) :
> A := Matrix([[0, 1, 2, 0], [2, 0, 7, 0], [6, 5, 0, 2], [1, 0, 4, 0]]);
```

$$A := \begin{bmatrix} 0 & 1 & 2 & 0 \\ 2 & 0 & 7 & 0 \\ 6 & 5 & 0 & 2 \\ 1 & 0 & 4 & 0 \end{bmatrix} \qquad (1)$$

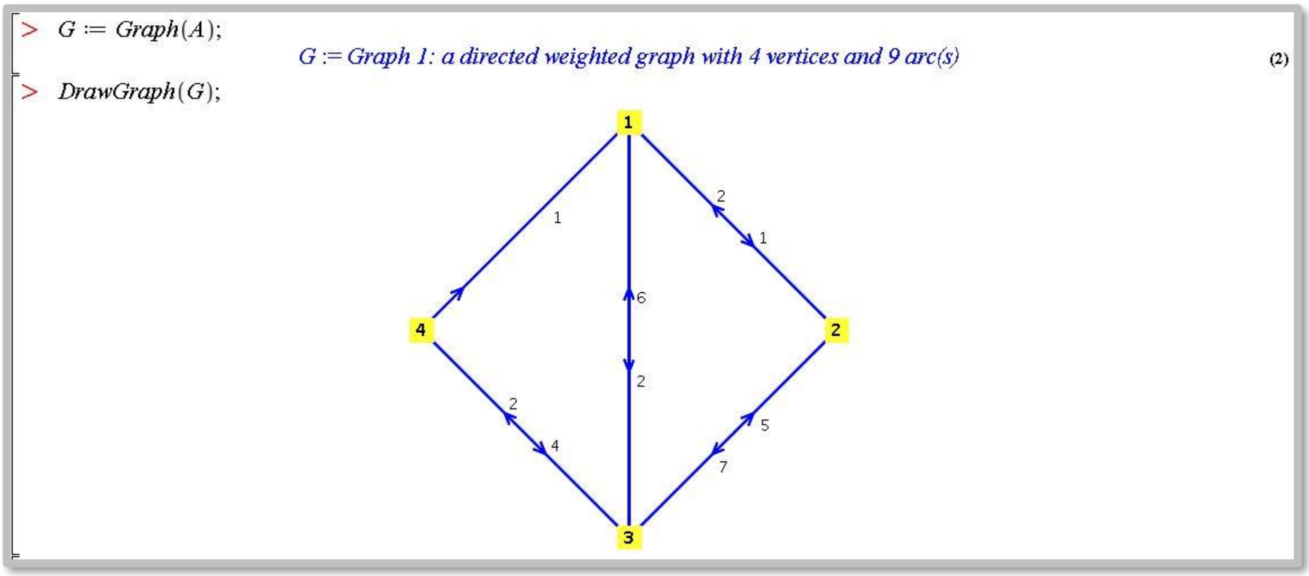


Рис. 7.

Для того щоб знайти степінь кожної вершини неорієнтованого графа, потрібно скористатись функцією

***Degree*(*G, i*)**

де ***i*** – номер вершини.

Задачу комівояжера, як задачу пошуку у неорієнтованому зваженому графі циклу, що проходить через всі вершини графа, крім першої, рівно по одному разу (такий цикл називають Гамільтоновим) розв’язуємо за допомогою функції

***TravelingSalesman*(*G*)**

Приклад. Відповідно до матриці відстаней (див. табл. 1), знайти кільцевий маршрут постачання готової продукції від заводу до чотирьох баз призначення з мінімізацією загальної відстані пробігу автотранспорту.

Таблиця 1.

C	**1**	**2**	**3**	**4**	**5**
1	∞	7	16	21	2
2	13	∞	21	15	43
3	25	3	∞	31	17
4	13	10	27	∞	33
5	9	2	19	14	∞

Розв’язання.

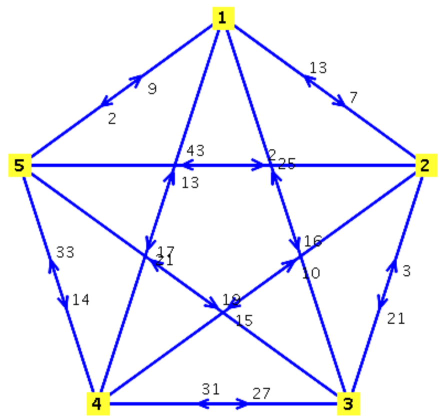
```
> restart;
> with(GraphTheory) :
```

Задаємо матрицю відстаней

$$> A := \begin{bmatrix} 0 & 7 & 16 & 21 & 2 \\ 13 & 0 & 21 & 15 & 43 \\ 25 & 3 & 0 & 31 & 17 \\ 13 & 10 & 27 & 0 & 33 \\ 9 & 2 & 19 & 14 & 0 \end{bmatrix};$$

$$A := \begin{bmatrix} 0 & 7 & 16 & 21 & 2 \\ 13 & 0 & 21 & 15 & 43 \\ 25 & 3 & 0 & 31 & 17 \\ 13 & 10 & 27 & 0 & 33 \\ 9 & 2 & 19 & 14 & 0 \end{bmatrix} \quad (1)$$

```
> G := Graph(A) : DrawGraph(G, style = circle);
```




```
> w, tour := TravelingSalesman(G); tour;
```

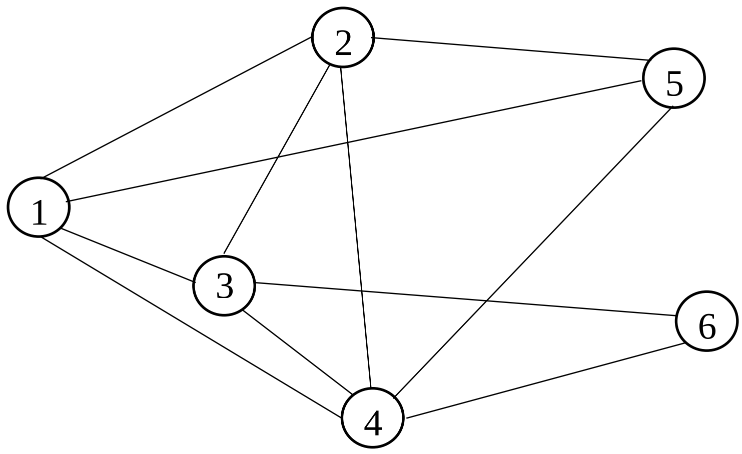
$$w, tour := 52, [1, 5, 3, 2, 4, 1]$$

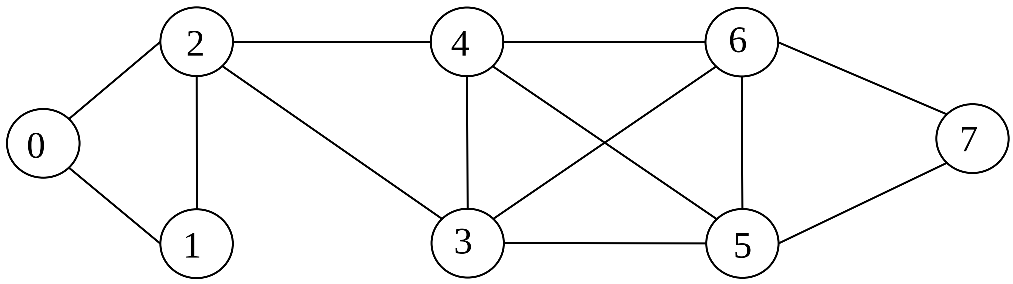
$$[1, 5, 3, 2, 4, 1] \quad (3)$$

Таким чином, отримано кільцевий маршрут 1→5→3→2→4→1 мінімальної довжини *w* = 52, що проходить через всі вершини рівно по одному разу.

Завдання для самостійного виконання.

1. Для заданих неорієнтованих графів
   а) знайти степінь кожної вершини;
   б) записати матрицю суміжності;
   в) знайти Ейлерів цикл (шлях) або довести, що він не існує.





2. Неорієнтований граф задано матрицею суміжностей.
Потрібно за допомогою пакету Maple:
а) побудувати граф, заданий матрицею суміжностей;
б) знайти степені всіх вершин графа;
в) побудувати Гамільтонів цикл.
Побудувати Ейлерів цикл (шлях) або довести, що він не існує.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0	1	0	0	0	1	0	0	0	0
2	1	0	1	1	0	0	1	1	1	0
3	0	1	0	1	1	0	1	1	0	0
4	0	1	1	0	1	1	0	0	1	0
5	0	0	1	1	0	1	0	0	0	1
6	1	0	0	1	1	0	1	0	1	1
7	0	1	1	0	0	1	0	1	0	0
8	0	1	1	0	0	0	1	0	1	0
9	0	1	0	1	0	1	0	1	0	0
10	0	0	0	0	1	1	0	0	0	0

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0	1	1	0	0	1	0	0	1	0
2	1	0	1	1	0	0	1	1	1	0
3	1	1	0	1	1	0	1	1	0	0
4	0	1	1	0	1	0	0	0	1	0
5	0	0	1	1	0	1	1	0	0	0
6	1	0	0	0	1	0	1	0	1	0
7	0	1	1	0	1	1	0	1	0	1
8	0	1	1	0	0	0	1	0	0	1
9	1	1	0	1	0	1	0	0	0	0
10	0	0	0	0	0	0	1	1	0	0