Лекція І І.
Пошук найкоротшого шляху в графі

§1 Постановка задачі

Задача про найкоротший шлях полягає у знаходженні найкоротшого шляху від заданої початкової вершини *a* до заданої вершини *z*.

Наступні дві задачі є безпосередніми узагальненнями сформульованої задачі про найкоротший шлях.

1. Для заданої початкової вершини знайти найкоротші шляхи від *a* до всіх інших вершин.
2. Знайти найкоротші шляхи між всіма парами вершин.

Розглянемо два алгоритми. Перший алгоритм розв'язує задачу 1, другий - спеціально призначений для розв'язування задачі 2.

§2 Алгоритм Дейкстри

Найефективнішим алгоритмом знаходження довжини найкоротшого шляху від фіксованої вершини до будь-якої іншої є алгоритм, запропонований 1959 р. нідерландським математиком Е. Дейкстрою (E. Dijkstra).

Цей алгоритм застосовується лише у випадку, коли **вага кожної дуги додатня**.

Нехай G=(V,E) – орієнтований граф, $w(v_i, v_j)$ – вага дуги (v_i, v_j) .

Пошук мінімального шляху здійснюється за допомогою присвоювання вершинам міток. Мітки є двох типів - тимчасові й постійні. Вершини з постійними мітками групують у множину M, яку називають *множиною позначених вершин*. Решта вершин має тимчасові мітки, і множину таких вершин позначають через T (T=V\M).

Величина постійної мітки вершини l(v) дорівнює довжині найкоротшого шляху від вершини *a* до вершини *v*. Якщо ж мітка тимчасова, то вона дорівнює довжині найкоротшого шляху, який проходить лише через вершини з постійними мітками.

Формальний опис алгоритму Дейкстри:

Крок 1. *Присвоювання початкових значень*. Виконати $l(a)=0$ і вважати цю мітку постійною. Виконати $l(v)=\infty$ для всіх $v \neq a$ і вважати ці мітки тимчасовими. Виконати $x=a$, $M=\{a\}$.

Крок 2. *Оновлення міток*. Для кожної вершини $v \in V \setminus M$ замінити мітки:

$$l(v)=\min\{ l(v), l(x)+w(x,v) \},$$

тобто оновлювати тимчасові мітки вершин, у які з вершини x іде дуга.

Крок 3. *Перетворення мітки у постійну.* Серед усіх вершин з тимчасовими мітками знайти вершину з мінімальною міткою, тобто знайти вершину *v** з умови

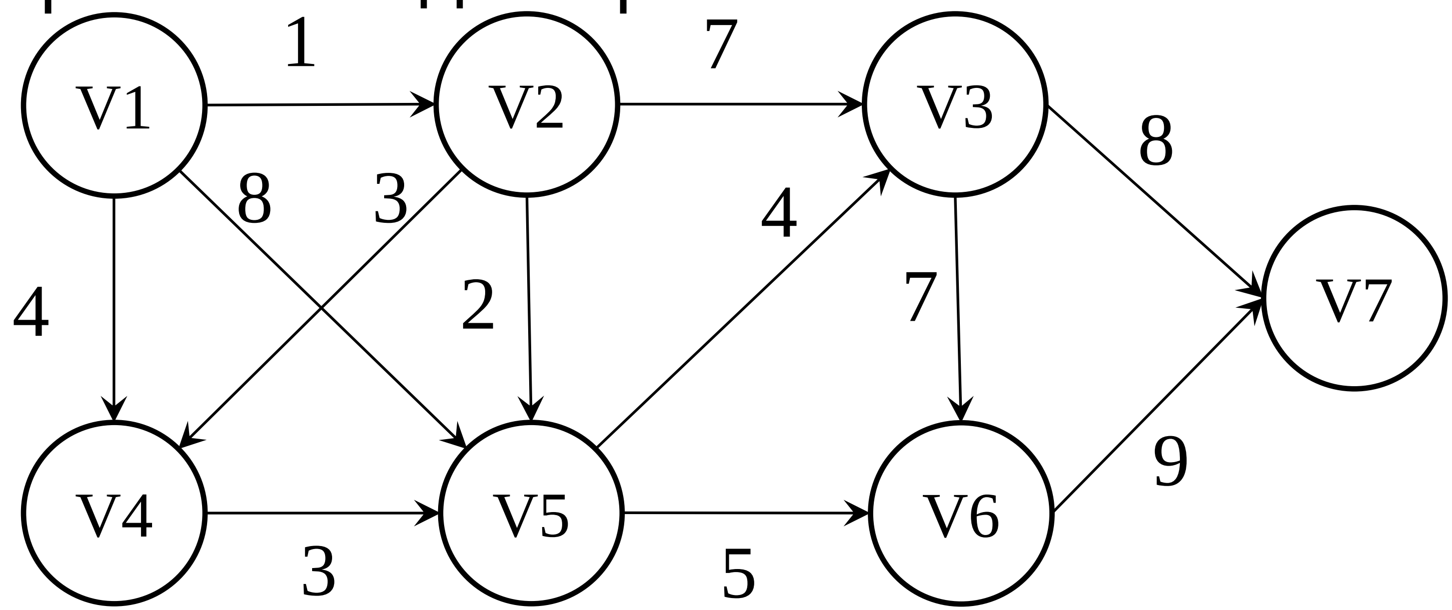
$$l(v^*)=\min l(v), v\in T, \quad T=V\backslash M.$$

Крок 4. Вважати мітку вершини *v** постійною і покласти $M=M\cup v^*$, $x=v^*$.

Крок 5. (*а*) (Якщо потрібно знайти шлях від *a* до z.) Якщо , x=z, то $l(z)$ – довжина найкоротшого шляху від *a* до *z*; зупинитись. Якщо $x\neq z$, то перейти до кроку 2.

(*б*) (Якщо потрібно знайти шлях від *a* до всіх інших вершин.) Якщо всі вершини отримали постійні мітки (включені у множину M), то ці мітки дають довжини найкоротших шляхів; зупинитись. Якщо деякі вершини мають тимчасові мітки, то перейти до кроку 2.

Приклад. Знайти найкоротший шлях від вершини V1 до вершини V7.



Початкові значення:

M={V1}, T=∅, *l*(V1)=0, *l*(V2)= *l*(V3)=… *l*(V7)=∞

Крок 1. T1= {V2(1, V1),V4(4,V1),V5(8,V1)} – min V2

M1= {V1, V2(1, V1)}

Крок 2. T2= {V4(4,V1),V5(8,V1),V4(4,V2),V5(3,V2), V3(8,V2)}= {V4(4,V1),V5(3,V2),V3(8,V2)}– min V5

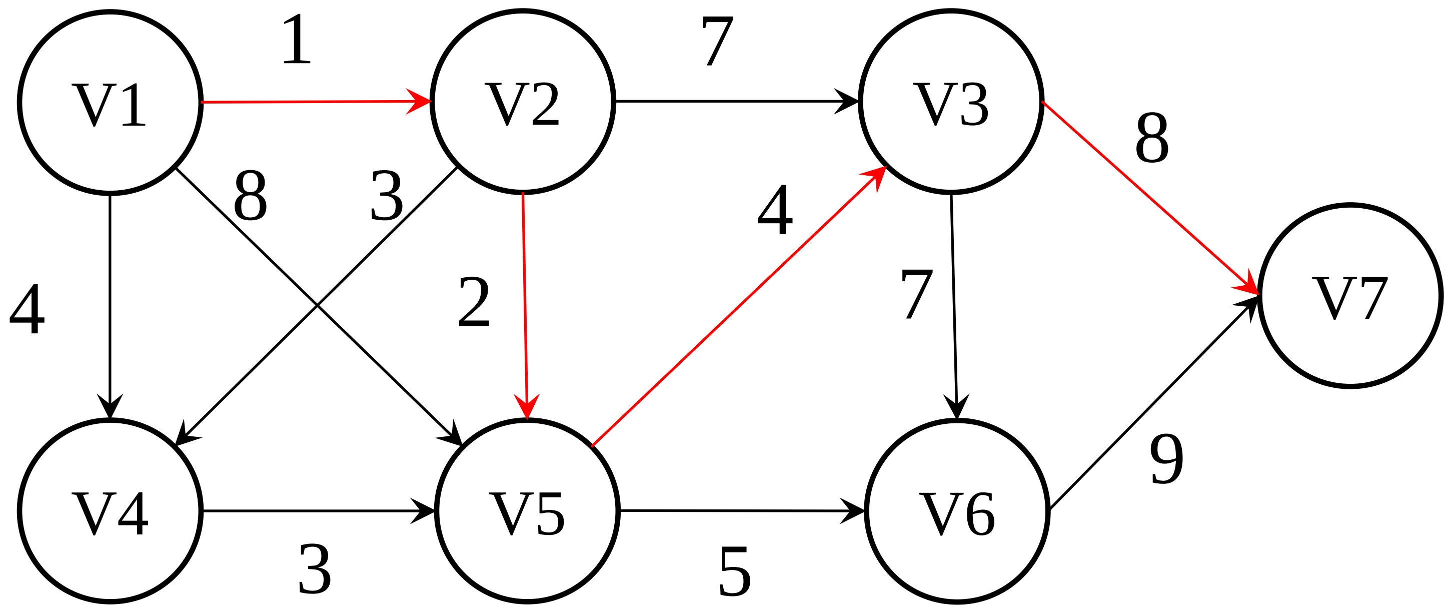
M2= {V1, V2(1, V1),V5(3,V2)}

Крок 3. $T_3 = \{V4(4,V_1), V3(8,V_2), V3(7,V_5), V6(8,V_5)\} =$
$= \{V4(4,V_1), V3(7,V_5), V6(8,V_5)\}$ – min V4
$M_3 = \{V_1, V_2(1, V_1), V5(3,V_2), V4(4,V_1)\}$

Крок 4. $T_4 = \{V3(7,V_5), V6(8,V_5)\}$ – min V3
$M_4 = \{V_1, V_2(1, V_1), V5(3,V_2), V4(4,V_1), V3(7,V_5)\}$

Крок 5. $T_5 = \{V6(8,V_5), V6(14,V_3), V7(15,V_3)\} =$
$= \{V6(8,V_5), V7(15,V_3)\}$ – min V6
$M_5 = \{V_1, V_2(1, V_1), V5(3,V_2), V4(4,V_1), V3(7,V_5), V6(8,V_5)\}$

Крок 6. $T6 = \{V7(15,V_3), V7(17,V6)\} = \{V7(15,V_3)\}$
$M6 = \{V_1, V_2(1, V_1), V5(3,V_2), V4(4,V_1), V3(7,V_5), V6(8,V_5), V7(15,V_3)\}$


V1
V2
V3
V4
V5
V6
V7
1
7
8
8
3
4
4
2
7
9
3
5

§3 Алгоритм Флойда

Алгоритм Флойда – алгоритм пошуку в графі найкоротших шляхів між кожною парою вершин.

В алгоритмі Флойда для довжин дуг дозволені від'ємні значення, проте не дозволена наявність циклів від'ємної довжини.

Використовує матрицю суміжності ваг та матрицю маршрутів.

Ідея алгоритму Флойда:

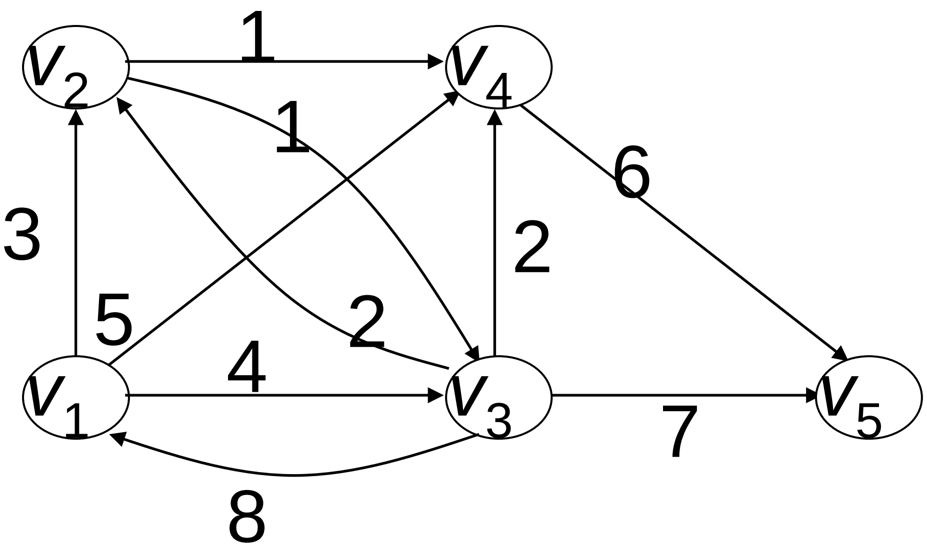
Припустимо, що нам відомі:

1) найкоротший шлях з вершини *i* у вершину k, в якому як внутрішні допускають використання лише перших (k-1) вершин;
2) найкоротший шлях з вершини k у вершину *j*, у якому як внутрішні допускають використання лише перших (k-1) вершин;
3) найкоротший шлях з вершини *i* у вершину *j*, у якому як внутрішні допускають використання лише перших (k-1) вершин.

Оскільки за припущенням граф G не містить циклів від'ємної довжини, то один з двох шляхів – шлях 3) або об'єднання шляхів 1) та 2) – є найкоротшим шляхом з вершини i у вершину j, у якому як внутрішні допускають використання лише перших k вершин.

$$w_{ij}^{(k)} = \min\left\{w_{ik}^{(k-1)} + w_{kj}^{(k-1)}, w_{ij}^{(k-1)}\right\}$$

Приклад. Знайти найкоротший шлях між всіма парами вершин.



Матриця суміжності ваг

$$
\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}
\begin{array}{c}
\begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\
\begin{pmatrix}
0 & 3 & 4 & 5 & \infty \\
\infty & 0 & 1 & 1 & \infty \\
8 & 2 & 0 & 2 & 7 \\
\infty & \infty & \infty & 0 & 6 \\
\infty & \infty & \infty & \infty & 0
\end{pmatrix}
\end{array}
\quad k=1
$$

Матриця маршруів

$$
\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}
\begin{array}{c}
\begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 \\
2 & 2 & 2 & 2 & 2 \\
3 & 3 & 3 & 3 & 3 \\
4 & 4 & 4 & 4 & 4 \\
5 & 5 & 5 & 5 & 5
\end{pmatrix}
\end{array}
$$

$$\begin{pmatrix} 0 & 3 & 4 & 5 & \infty \\ \infty & 0 & 1 & 1 & \infty \\ 8 & 2 & 0 & 2 & 7 \\ \infty & \infty & \infty & 0 & 6 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \quad k=2 \quad \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 3 & 4 & (4) & \infty \\ \infty & 0 & 1 & 1 & \infty \\ 8 & 2 & 0 & 2 & \infty \\ \infty & \infty & \infty & 0 & 6 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \quad k=3 \quad \begin{pmatrix} 1 & 1 & 1 & (2) & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{pmatrix}$$

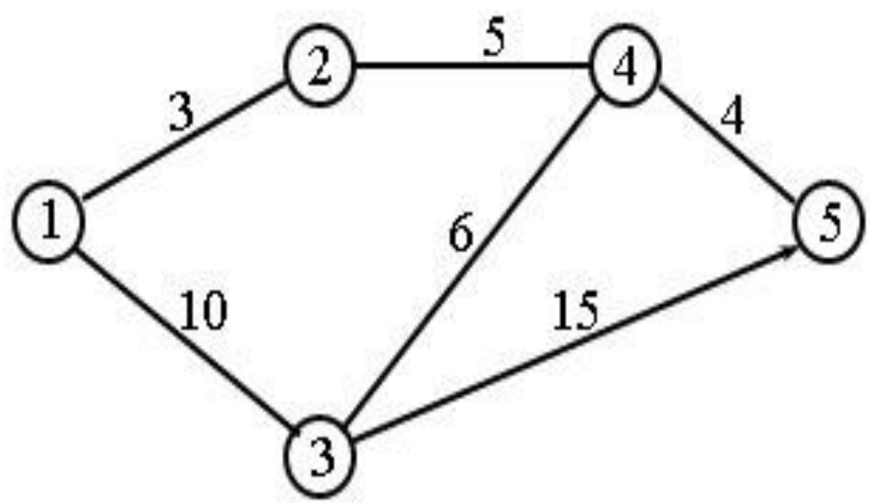
$$\begin{pmatrix} 0 & 3 & 4 & 4 & \infty \\ (9) & 0 & 1 & 1 & \infty \\ 8 & 2 & 0 & 2 & \infty \\ \infty & \infty & \infty & 0 & 6 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \quad k=4 \quad \begin{pmatrix} 1 & 1 & 1 & 2 & 1 \\ (3) & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 3 & 4 & 4 & (10) \\ 9 & 0 & 1 & 1 & (7) \\ 8 & 2 & 0 & 2 & (8) \\ \infty & \infty & \infty & 0 & 6 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \quad k = 5 \quad \begin{pmatrix} 1 & 1 & 1 & 2 & (4) \\ 3 & 2 & 2 & 2 & (4) \\ 3 & 3 & 3 & 3 & (4) \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{pmatrix}$$

Відповідь:

$$\begin{pmatrix} 0 & 3 & 4 & 4 & 10 \\ 9 & 0 & 1 & 1 & 7 \\ 8 & 2 & 0 & 2 & 8 \\ \infty & \infty & \infty & 0 & 6 \\ \infty & \infty & \infty & \infty & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 1 & 2 & 4 \\ 3 & 2 & 2 & 2 & 4 \\ 3 & 3 & 3 & 3 & 4 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{pmatrix}$$

Знайдемо для мережі найкоротші шляхи між будь-якими двома вузлами. Відстань між вузлами цієї мережі проставлена біля відповідних ребер. Ребро (3, 5) орієнтоване, тому не допускається рух від вузла 5 до вузла 3. Всі інші ребра допускають рух в обидві сторони:



Крок 0. Початкові матриці $\mathbf{D_0}$ і $\mathbf{S_0}$ будуються безпосередньо за заданою схемою мережі. Матриця $\mathbf{D_0}$ симетрична, за виключенням пари елементів $\mathbf{d_{35}}$ і $\mathbf{d_{53}}$, де $\mathbf{d_{53}}$ дорівнює безкінечності, оскільки неможливий перехід від вузла 5 до вузла 3:

D_0

	1	2	3	4	5
1	—	3	10	∞	∞
2	3	—	∞	5	∞
3	10	∞	—	6	15
4	∞	5	6	—	4
5	∞	∞	∞	4	—

S_0

	1	2	3	4	5
1	—	2	3	4	5
2	1	—	3	4	5
3	1	2	—	4	5
4	1	2	3	—	5
5	1	2	3	4	—

Початковий стан

<u>Крок1</u>. В матриці $\mathbf{D_0}$ виділені провідні рядок і стовпчик **(k = 1)**. Подвійною рамкою представлені елементи $\mathbf{d_{23}}$ і $\mathbf{d_{32}}$, єдині серед елементів матриці $\mathbf{D_0}$, значення яких можна покращати за допомогою трикутного оператора. Таким чином, щоб на основі матриць $\mathbf{D_0}$ і $\mathbf{S_0}$ отримати матриці $\mathbf{D_1}$ і $\mathbf{S_1}$, виконуємо наступні дії:

1.Замінюємо $\mathbf{d_{23}}$ на $\mathbf{d_{21} + d_{13} = 3 + 10 = 13}$ і встановлюємо $\mathbf{s_{23} = 1}$.

2.Замінюємо $\mathbf{d_{32}}$ на $\mathbf{d_{31} + d_{12} = 10 + 3 = 13}$ і встановлюємо $\mathbf{s_{32} = 1}$.

Матриці $\mathbf{D_1}$ і $\mathbf{S_1}$ мають наступний вигляд:

D_1

	1	2	3	4	5
1	—	3	10	∞	∞
2	3	—	**13**	5	∞
3	10	**13**	—	6	15
4	∞	5	6	—	4
5	∞	∞	∞	4	—

S_1

	1	2	3	4	5
1	—	2	3	4	5
2	1	—	**1**	4	5
3	1	**1**	—	4	5
4	1	2	3	—	5
5	1	2	3	4	—

Матриці $\mathbf{D_1}$ і $\mathbf{S_1}$

<u>Крок2</u>. Покладаємо **k = 2**; в матриці $\mathbf{D_1}$ виділені провідні рядок і стовпчик. Трикутний оператор застосовується до елементів матриці $\mathbf{D_1}$ і $\mathbf{S_1}$, що виділені подвійною рамкою. В результаті отримуємо матриці $\mathbf{D_2}$ і $\mathbf{S_2}$:

D_2

	1	2	3	4	5
1	—	3	10	8	∞
2	3	—	13	5	∞
3	10	13	—	6	15
4	8	5	6	—	4
5	∞	∞	∞	4	—

S_2

	1	2	3	4	5
1	—	2	3	2	5
2	1	—	1	4	5
3	1	1	—	4	5
4	2	2	3	—	5
5	1	2	3	4	—

Матриці $\mathbf{D_2}$ і $\mathbf{S_2}$

Крок **_3_.** Покладаємо **k = 3**; в матриці $\mathbf{D_2}$ виділені провідні рядок і стовпчик. Трикутний оператор застосовується до елементів матриці $\mathbf{D_2}$ і $\mathbf{S_2}$, що виділені подвійною рамкою. В результаті отримуємо матриці $\mathbf{D_3}$ і $\mathbf{S_3}$:

D_3

	1	2	3	4	5
1	—	3	10	8	25
2	3	—	13	5	28
3	10	13	—	6	15
4	8	5	6	—	4
5	∞	∞	∞	4	—

S_3

	1	2	3	4	5
1	—	2	3	2	3
2	1	—	1	4	3
3	1	1	—	4	5
4	2	2	3	—	5
5	1	2	3	4	—

Матриці $\mathbf{D_3}$ і $\mathbf{S_3}$

Крок4. Покладаємо **k = 4**, провідні рядок і стовпчик в матриці $\mathbf{D_3}$ виділені. Отримуємо нові матриці $\mathbf{D_4}$ і $\mathbf{S_4}$:

D_4

	1	2	3	4	5
1	—	3	10	8	**12**
2	3	—	**11**	5	**9**
3	10	**11**	—	6	**10**
4	8	5	6	—	4
5	**12**	**9**	**10**	4	—

S_4

	1	2	3	4	5
1	—	2	3	2	**4**
2	1	—	**4**	4	**4**
3	1	**4**	—	4	**4**
4	**2**	2	3	—	5
5	**4**	**4**	**4**	4	—

Матриці $\mathbf{D_4}$ і $\mathbf{S_4}$

Крок5. Покладаємо **k = 5**, провідні рядок і стовпчик в матриці $\mathbf{D_4}$ виділені. Жодних дій на цьому кроці не виконуємо; обчислення закінчені.

Кінцеві матриці $\mathbf{D_4}$ і $\mathbf{S_4}$ містять всю інформацію, що необхідна для визначення найкоротших шляхів між будь-якими двома вузлами мережі. Наприклад найкоротша відстань між вузлами 1 і 5 дорівнює $\mathbf{d_{15} = 12}$.