Навчальна дисципліна: **Дискретна математика**

Лектор:

професор Кучук Георгій Анатолійович

E-mail: kuchuk56@ukr.net

3 семестр навчання на бакалавраті

Наприкінці семестру - іспит

Тема 5. Графи

Лекція 5.3. Знаходження оптимальних маршрутів

Питання лекції

1. Постановка завдання мінімізації мережі.
2. Алгоритм Краскала.
3. Алгоритм Прима.
4. Алгоритм Прима з підмножинами вершин.
5. Постановка задачі знаходження мінімальної відстані в мережі.
6. Алгоритм Дейкстри для упорядкованого графа.
7. Алгоритм Дейкстри для будь-якого графа.

Рекомендована література

1. *Конспект лекцій.URL: https://drive.google.com/drive/folders/12QYRD4L8kQr0g48DJVN386FrISDuyPwQ?usp=sharing*

2. Олійник Л.О. Дискретна математика: Навч. посібник. 2015. 256 с. URL:: https://www.dstu.dp.ua/Portal/Data/3/17/3-17-b2.pdf

3. Балога С.І. Дискретна математика*. Навчальний посібник. – Ужгород: ПП «АУТДОР-. ШАРК», 2021. – 124 с. https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/3415/1/%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE-%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%B8%D1%87%D0%BD%D0%B8%D0%B9%20%D0%BF%D0%BE%D1%81%D1%96%D0%B1%D0%BD%D0%B8%D0%BA.pdf*

1. Постановка завдання мінімізації мережі

Загальне завдання. Між N віддаленими підмережами ($S_1, ..., S_N$) МСС необхідно прокласти оптоволоконний кабель.

Витрати на прокладку кабелю між підмережами S_i і S_j складають C_{ij} (умов. од.).

Всі підмережі рівноправні.

Мінімізувати витрати на прокладку кабелю.

Основний алгоритм – знаходження **мінімального кістякового дерева.**

2. Алгоритм Краскала

Алгоритм Краскала починається з побудови виродженого лісу, що містить V дерев, кожне з яких складається з однієї вершини.

Далі виконуються операції об'єднання двох дерев (після кроку 0 одне дерево – декілька зв'язаних вершин, друге – ізольована вершина), для чого використовуються найкоротші можливі ребра, поки не утвориться єдине дерево.

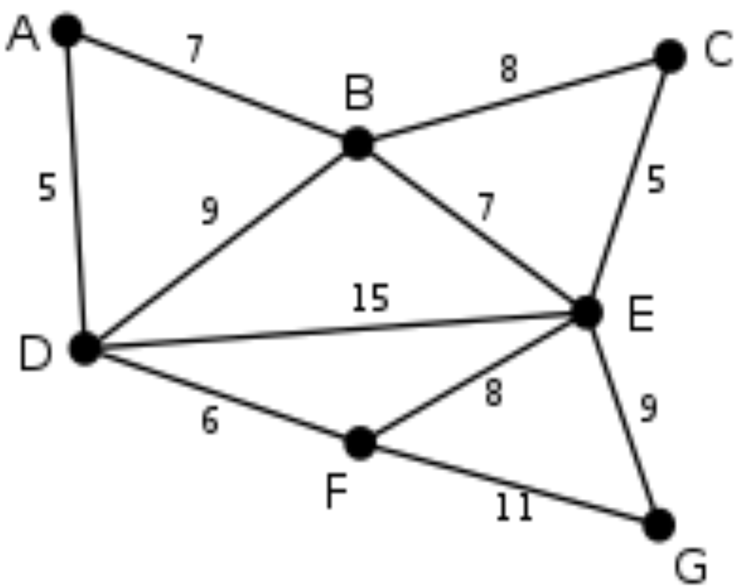
Ознака закінчення процесу – кількість доданих ребер дорівнює кількості вершин, зменшеної на одиницю.

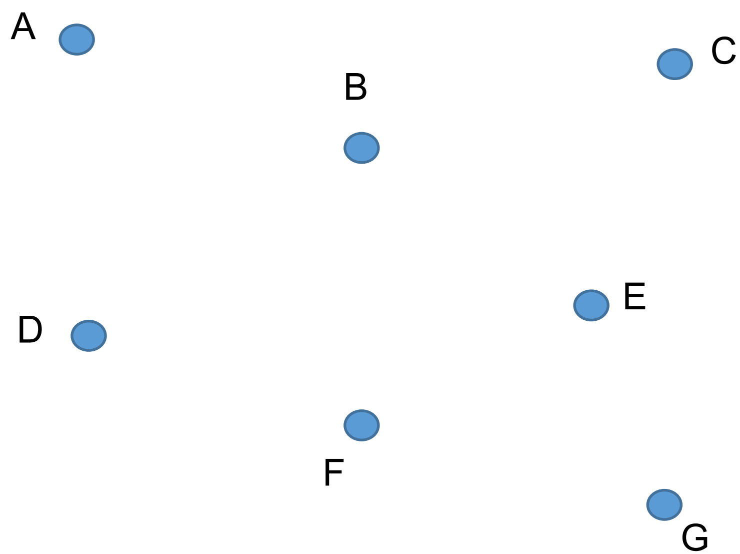
Це дерево і буде **мінімальним кістяковим деревом**.

Приклад. Розглянемо зважений неорієнтований (7,11)-граф, котрий задає всі можливі варіанти прокладення кабелю між 7 підмережами: A, B, C, D, E, F, G. Яка мінімальна довжина кабелю необхідна?

Крок 0. Вироджений ліс, що складається з 7 незв’язаних вершин.
Формуємо впорядкований за вагою список ребер.
Поточна довжина кабелю L = 0.

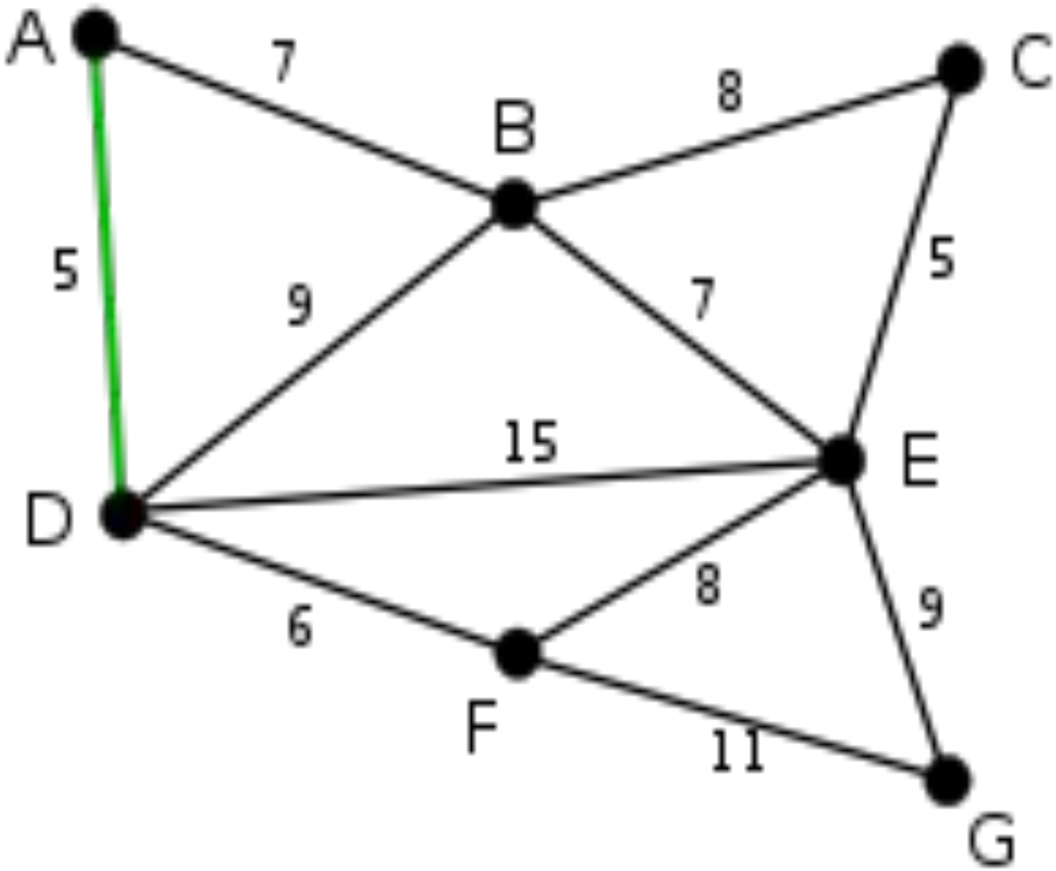
Початковий граф Граф кроку 0 Список ребер





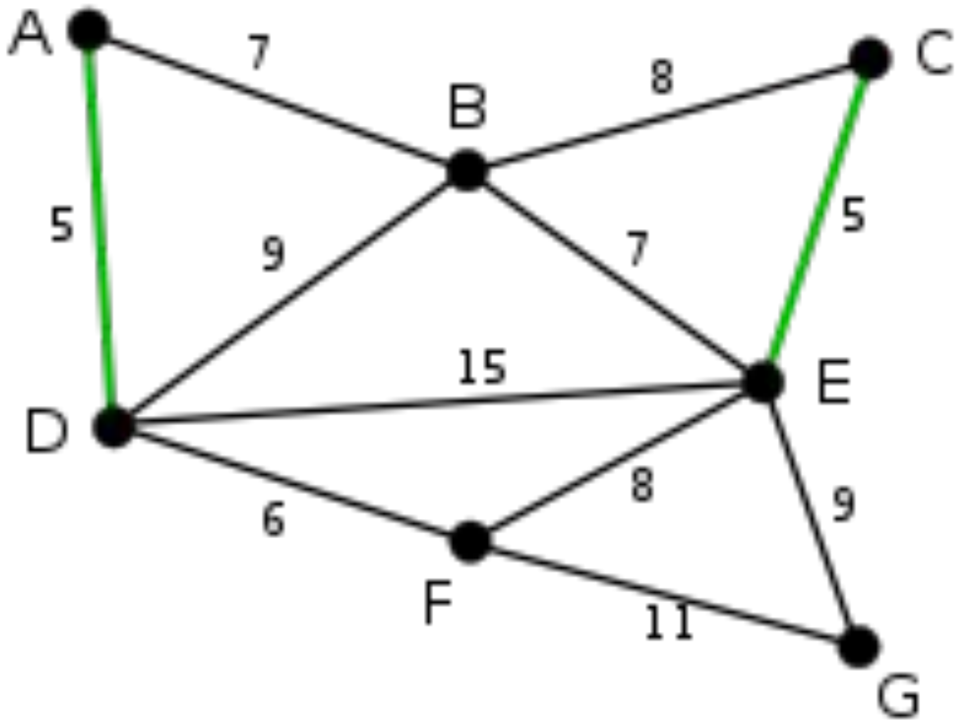
АД (5)
CE (5)
DF (6)
AB (7)
BE (7)
BC (8)
EF (8)
BD (9)
EG (9)
FG (11)
DE (15)

Крок 1. AD і CE мають найменшу вагу 5, і AD вибирається з них довільно та додається до кістякового дерева (зелений колір). L = 0 + 5 = 5.



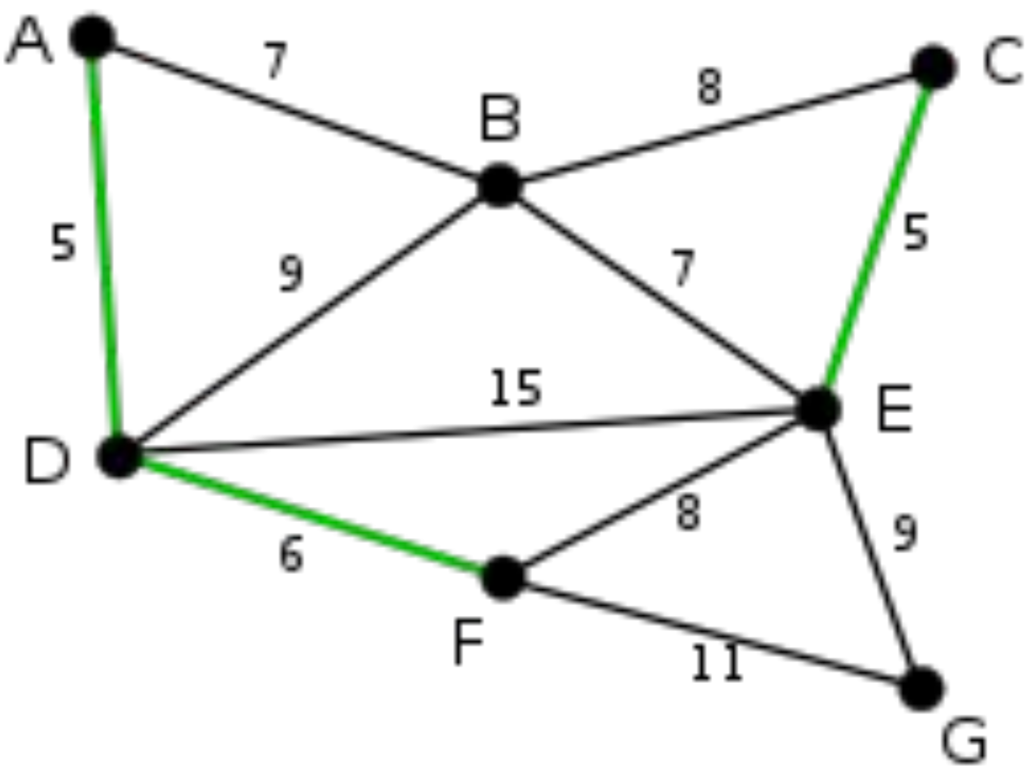
АД (5)
CE (5)
DF (6)
AB (7)
BE (7)
BC (8)
EF (8)
BD (9)
EG (9)
FG (11)
DE (15)

Крок 2. Далі CE є найлегшим ребром з вагою 5, тому воно також додається до дерева.
L = 5 + 5 = 10.



АД (5)
CE (5)
DF (6)
AB (7)
BE (7)
BC (8)
EF (8)
BD (9)
EG (9)
FG (11)
DE (15)

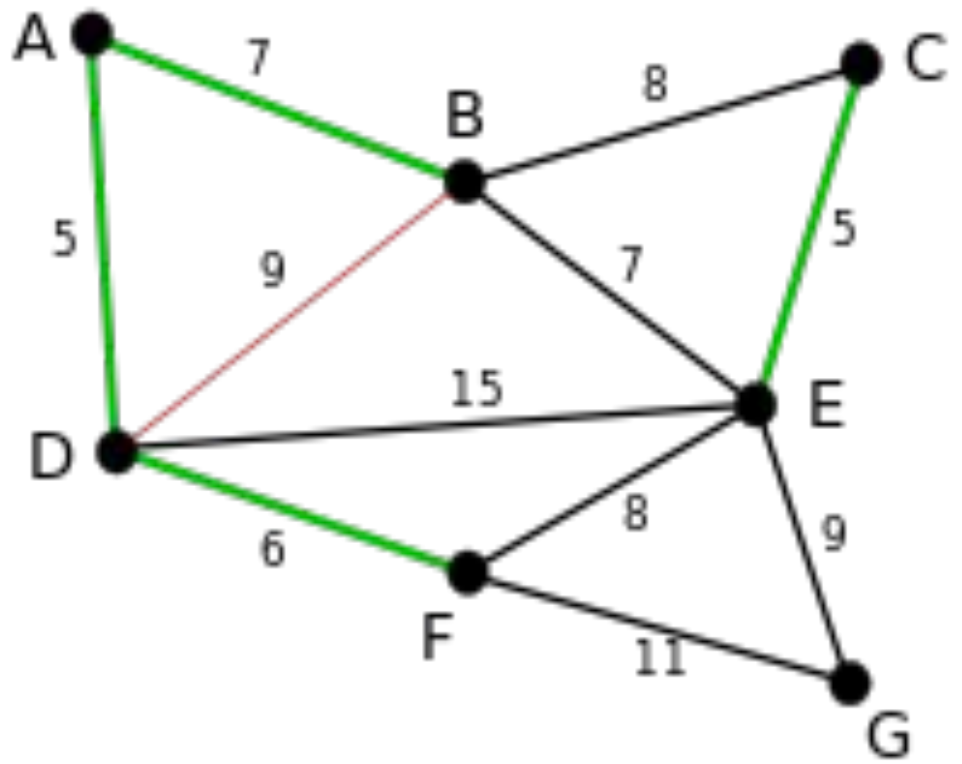
Крок 3. Аналогічним чином обирається найлегше з недоданих ребер графу DF з вагою 6 і додається до кістякового дерева.
L = 10 + 6 = 16.



АД (5)
CE (5)
DF (6)
AB (7)
BE (7)
BC (8)
EF (8)
BD (9)
EG (9)
FG (11)
DE (15)

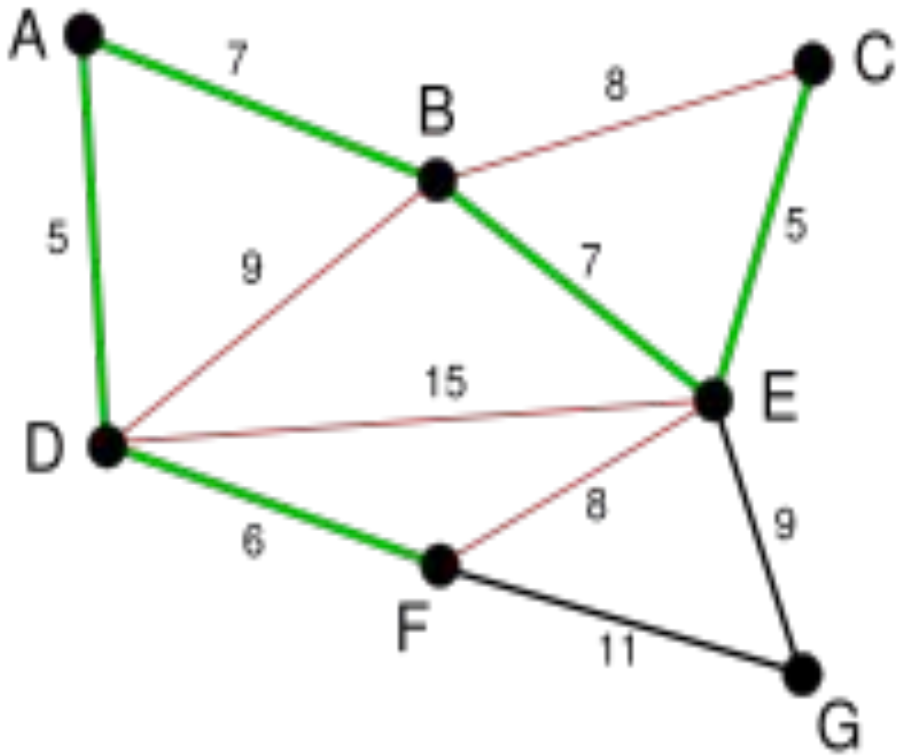
Крок 4. Наступними найлегшими ребрами є AB і BE, вагою 7. AB обирається довільно і додається до кістякового дерева. BD фарбується у червоний колір, оскільки воно є частиною циклу ABD.

L = 16 + 7 = 23.



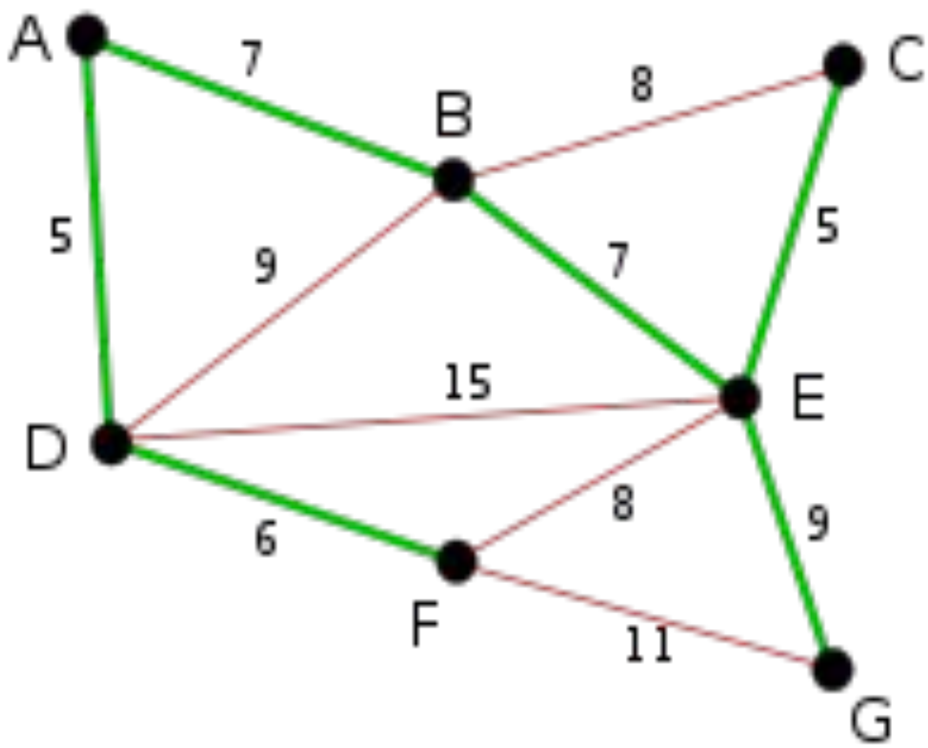
АД (5)
CE (5)
DF (6)
AB (7)
BE (7)
BC (8)
EF (8)
BD (9)
EG (9)
FG (11)
DE (15)

Крок 5. Наступним додається ребро ВЕ з вагою 7.
L = 23 + 7 = 30.
Червоним забарвлюємо ребра ВС (цикл ВСЕ), DE (цикл DEBA) і FE (цикл FEBAD).



АД (5)
CE (5)
DF (6)
AB (7)
BE (7)
BC (8)
EF (8)
BD (9)
EG (9)
FG (11)
DE (15)

Крок 6. Додаємо ребро EG вагою 9 і отримуємо мінімальне кістякове дерево (це шосте зелене ребро, тому процес закінчено),
L = 30 + 9 = 39.



АД (5)
CE (5)
DF (6)
AB (7)
BE (7)
BC (8)
EF (8)
BD (9)
EG (9)
FG (11)
DE (15)

Відповідь: мінімальна довжина кабелю складає **39** км, а на графі наведений один із можливих варіантів (у даному випадку єдиний).

5. Постановка задачі знаходження мінімальної відстані в мережі

Є N мережних вузлів, пронумерованих від 1 до N.

Необхідно передати повідомлення з вузла 1 до вузла N з мінімальними витратами мережного ресурсу, якщо витрати при передачі з вузла i у вузол j дорівнюють d_{ij} (i < j, якщо граф впорядкований).

Якщо граф впорядкований, то передбачається, що вузли впорядковані і повідомлення може бути передано далі тільки на вузол з великим номером (це можливо, оскільки передбачається мережа без циклів).

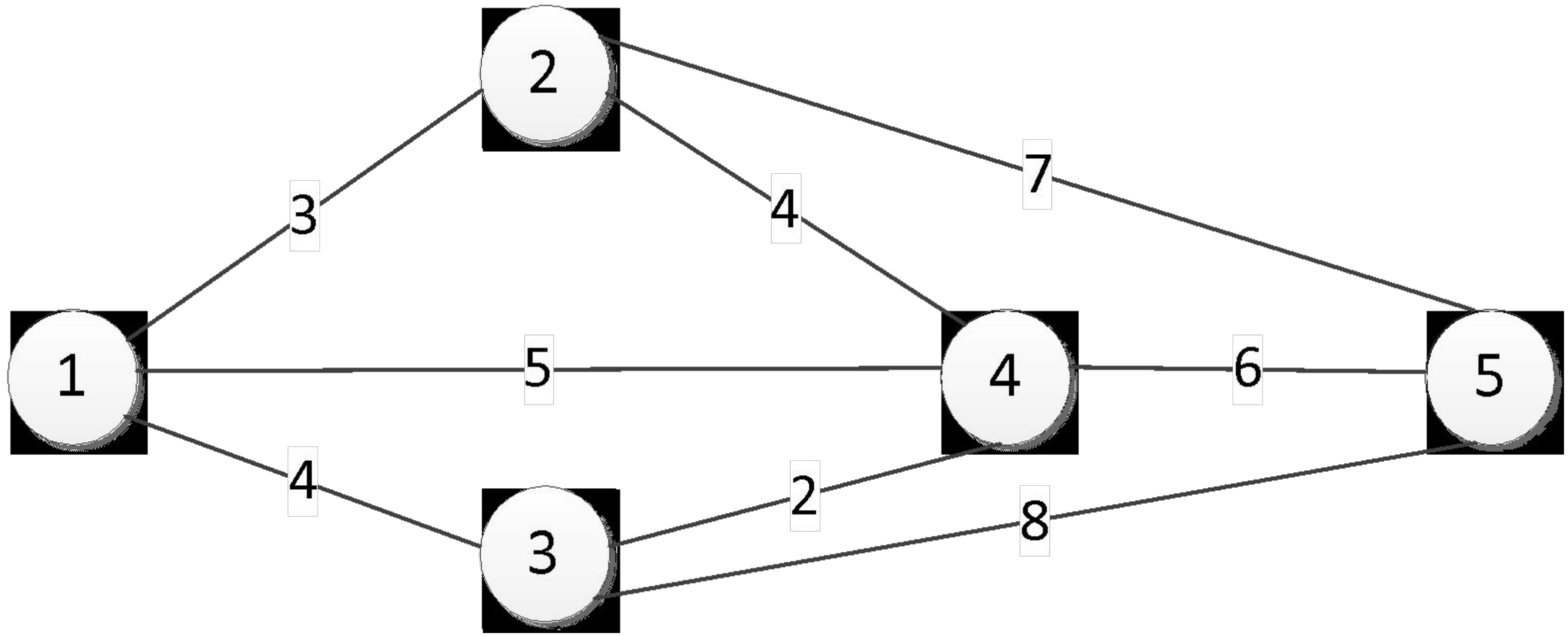
Основний алгоритм – **знаходження найкоротшого маршруту.**

6. Алгоритм Дейкстри для упорядкованого графа

Крок 1. Нехай u_j – довжина мінімального маршруту з вузла 1 у вузол j;
j=0; u_1=0.

Крок 2. j=j+1.
Якщо j > N, то u_N – довжина маршруту, котрий шукаємо.

Крок 3. $u_j = \min\limits_{i<j}\left\{u_i + d_{ij}\right\}$.



1) $u_1=0$

2) $u_2=\min\{u_1+d_{12}\}=3$

3) $u_3=\min\{u_1+d_{13}\}=4$

4) $u_4=\min\{u_1+d_{14};\ u_2+d_{24};\ u_3+d_{34};\}=\min\{0+5;\ 3+4;\ 4+2\}=5$

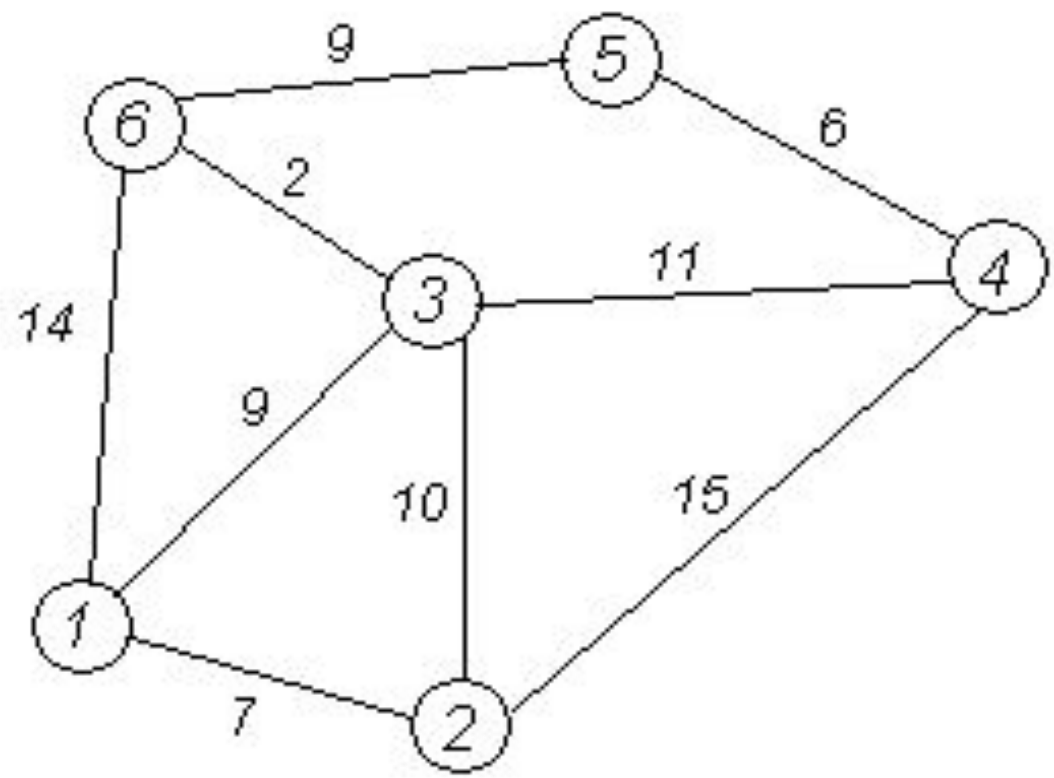
5) $u_5=\min\{u_2+d_{25};\ u_3+d_{35};\ u_4+d_{45};\}=\min\{3+7;\ 4+8;\ 5+6\}=10$

Шлях $5\rightarrow2\rightarrow1=10$ (од.)

7. Алгоритм Дейкстри для будь-якого графа

Дано неорієнтований зв'язний граф G(V, U). Знайти відстань від вершини a до всіх інших вершин V.

Зберігатимемо по-точну мінімальну відстань до всіх вершин V (від даної вершини a) і на кожному кроці алгоритму намагатимемося зменшити цю відстань.

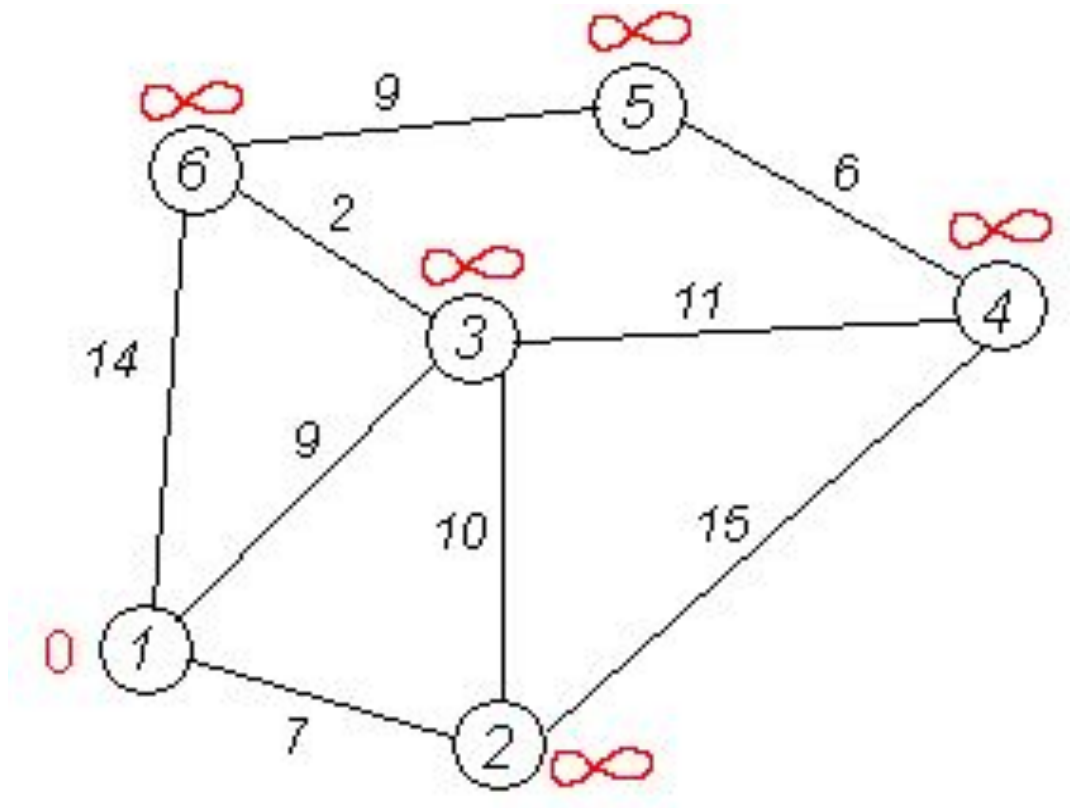


Спочатку встановимо відстані до всіх вершин рівними нескінченості, а до вершини a — нулю.

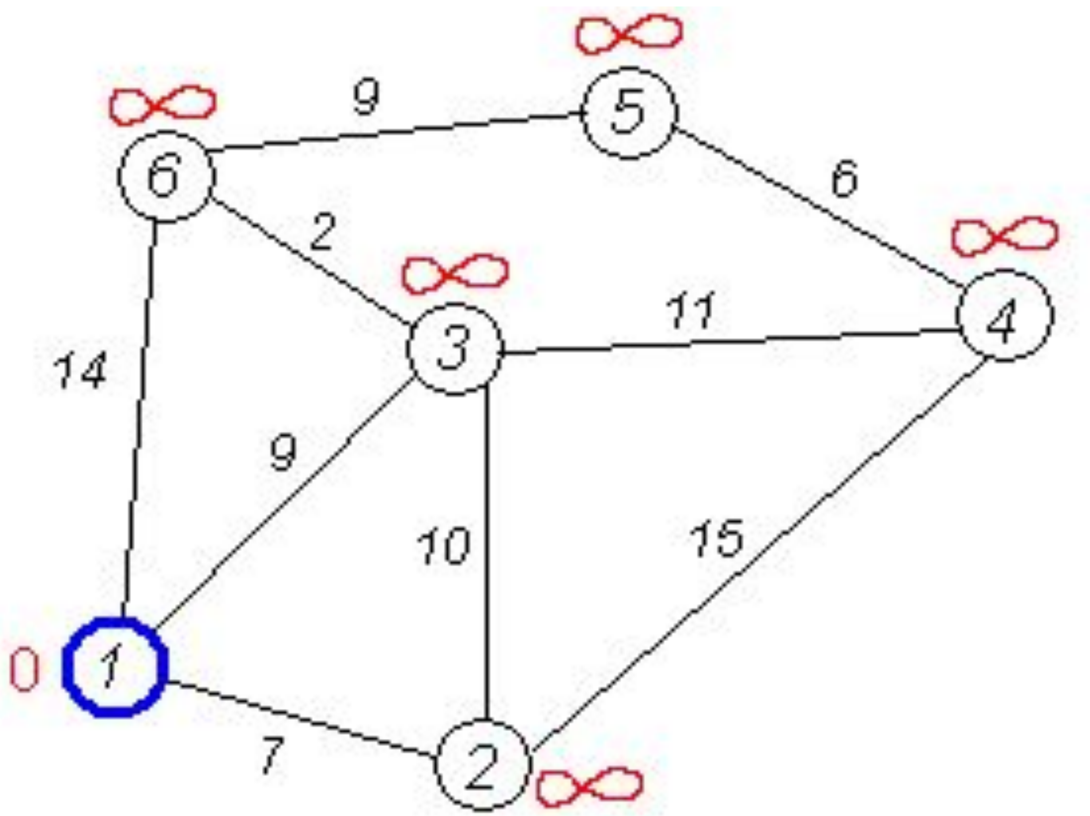
Покроковий приклад

Крок 1. Ініціалізація.

Встановлюємо відстані до всіх вершин рівними нескінченості, а до вершини a — нулю. Жодної вершини графу ще не опрацьовано.

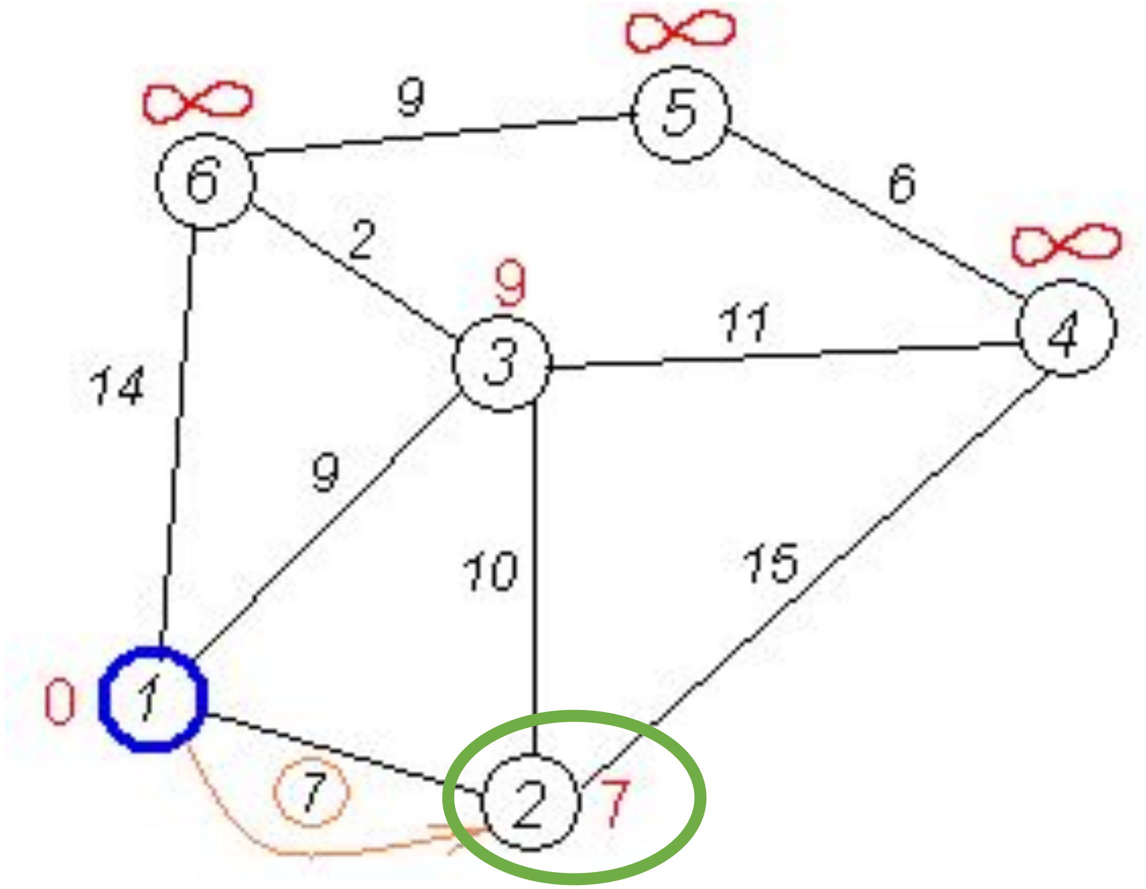


Крок 2. Початок опрацьовування начальної вершини.
Знаходимо таку вершину (із ще не опрацьованих), поточна найкоротша відстань до якої мінімальна. Це вершина **1**. Починаємо її опрацьовувати (відмічаємо синім кружечком). Для цього будемо обходити всіх її сусідів і, знаходимо найкоротший шлях до кожного із сусідів.



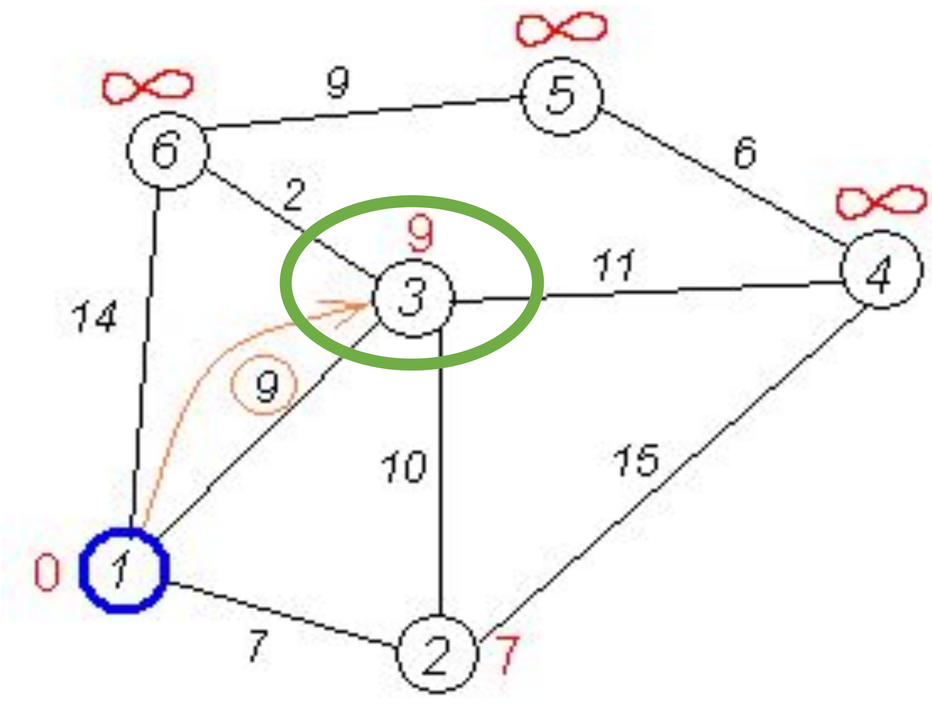
Крок 3. Шлях до першого сусіда

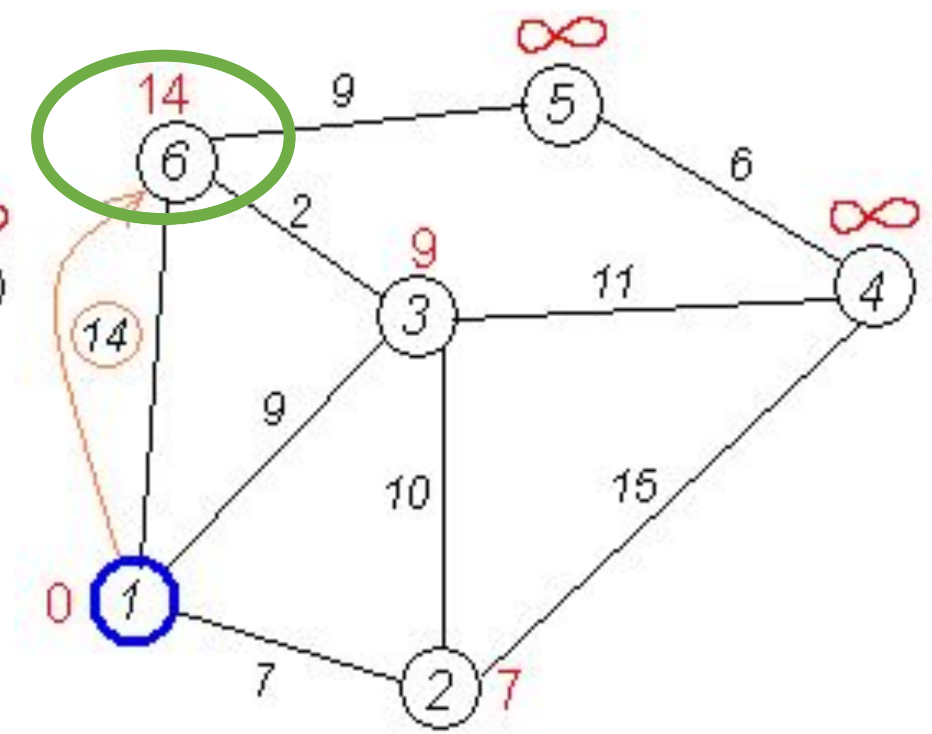
Перший по порядку сусід 1-ї вершини — 2-а вершина. Шлях до неї через 1-у вершину дорівнює найкоротшій відстані до 1-ї вершини + довжина дуги між 1-ю та 2-ю вершиною, тобто 0 + 7 = 7 (червона стрілочка). Це менше поточного найкоротшого шляху до 2-ї вершини, (нескінченості), тому найкоротший шлях до 2-ї вершини дорівнює 7.



Кроки 4, 5. Пошук шляхів до інших сусідів

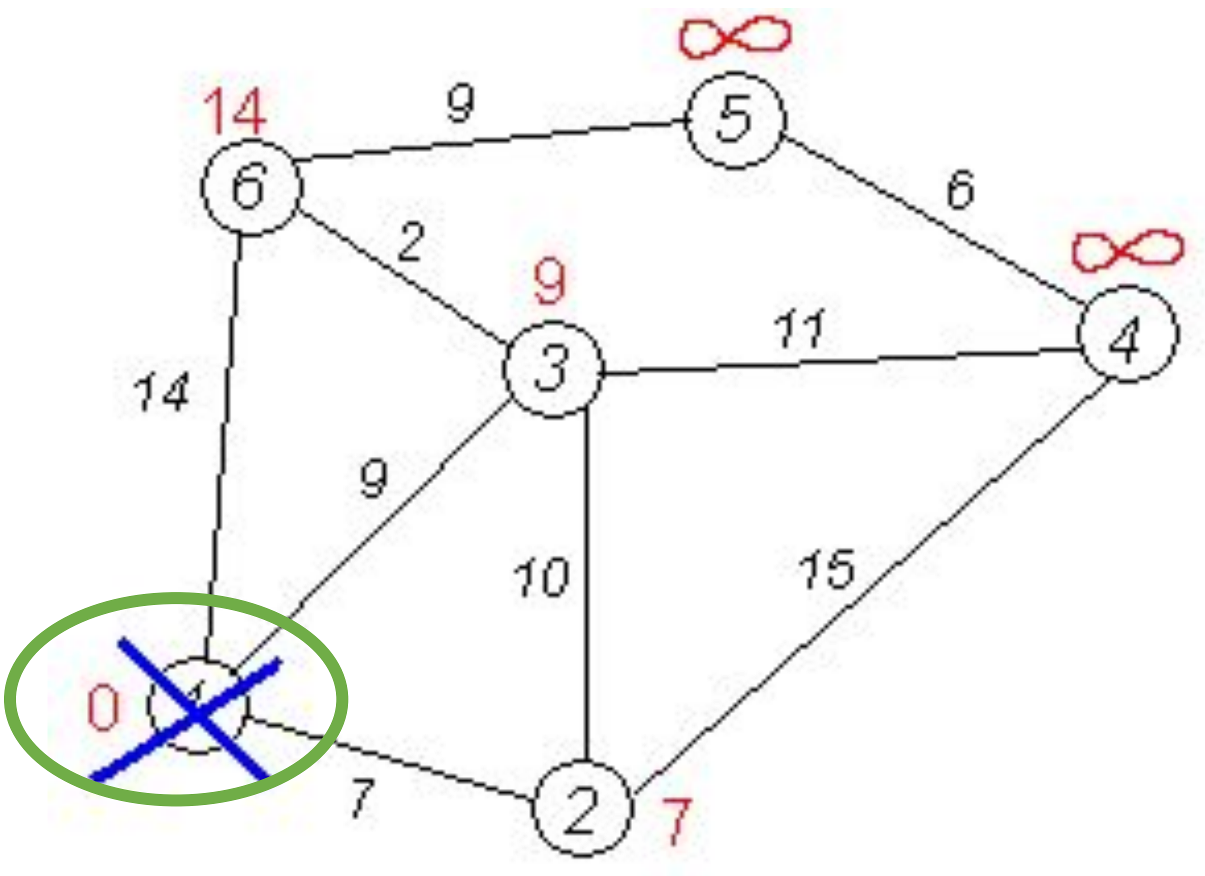
Аналогічну операцію пророблюємо з 2 іншими сусідами першої вершини — 3-ю та 6-ю вершинами.





Крок 6. Пошук шляхів до інших сусідів

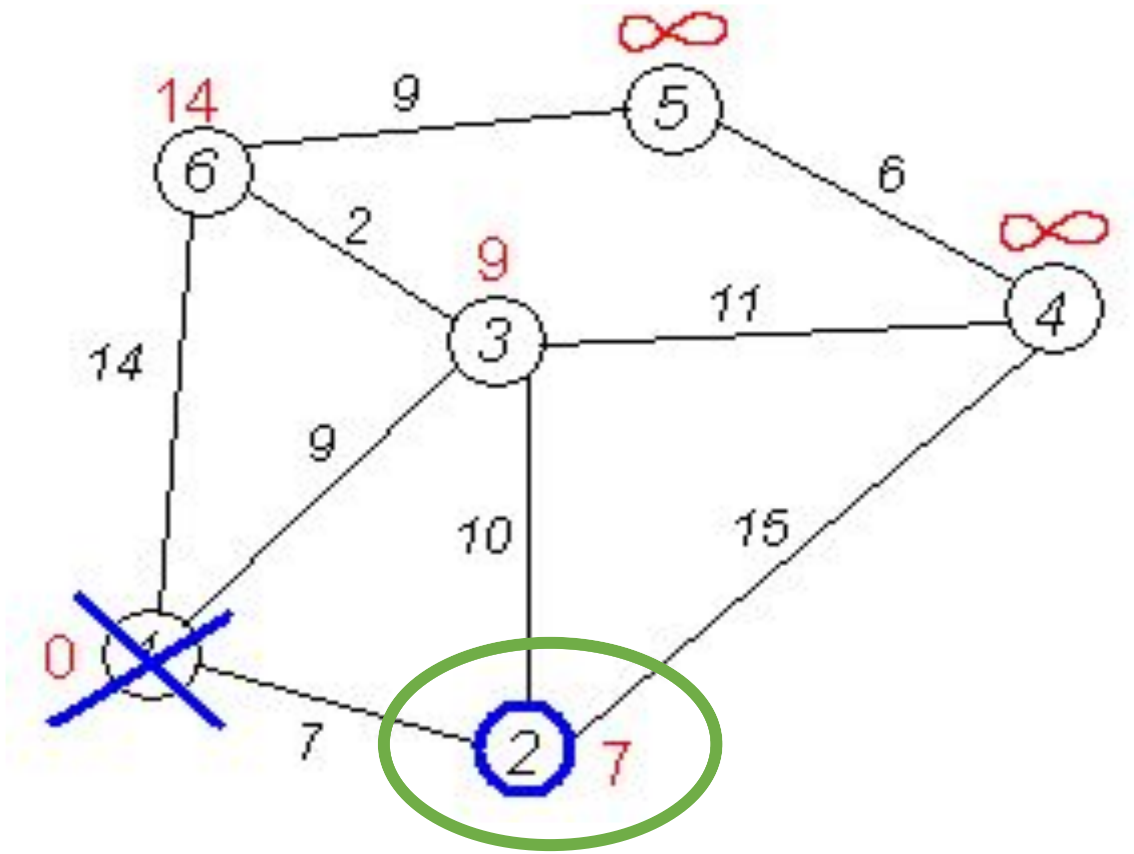
Всі сусіди вершини 1 перевірені. Поточна мінімальна відстань до вершини 1 вважається остаточною і обговоренню не підлягає (те, що це дійсно так, вперше довів Дейкстра). Тому викреслимо її з графу, щоб відмітити цей факт.



Крок 7 (2). Пошук найближчої необробленої вершини

Практично відбувається повернення до кроку 2. Знову знаходимо «найближчу» необроблену (невикреслену) вершину. Це вершина 2 з поточною найкоротшою відстанню до неї = 7. Обводимо її синім кружечком і починаємо опрацьовувати її сусідів.

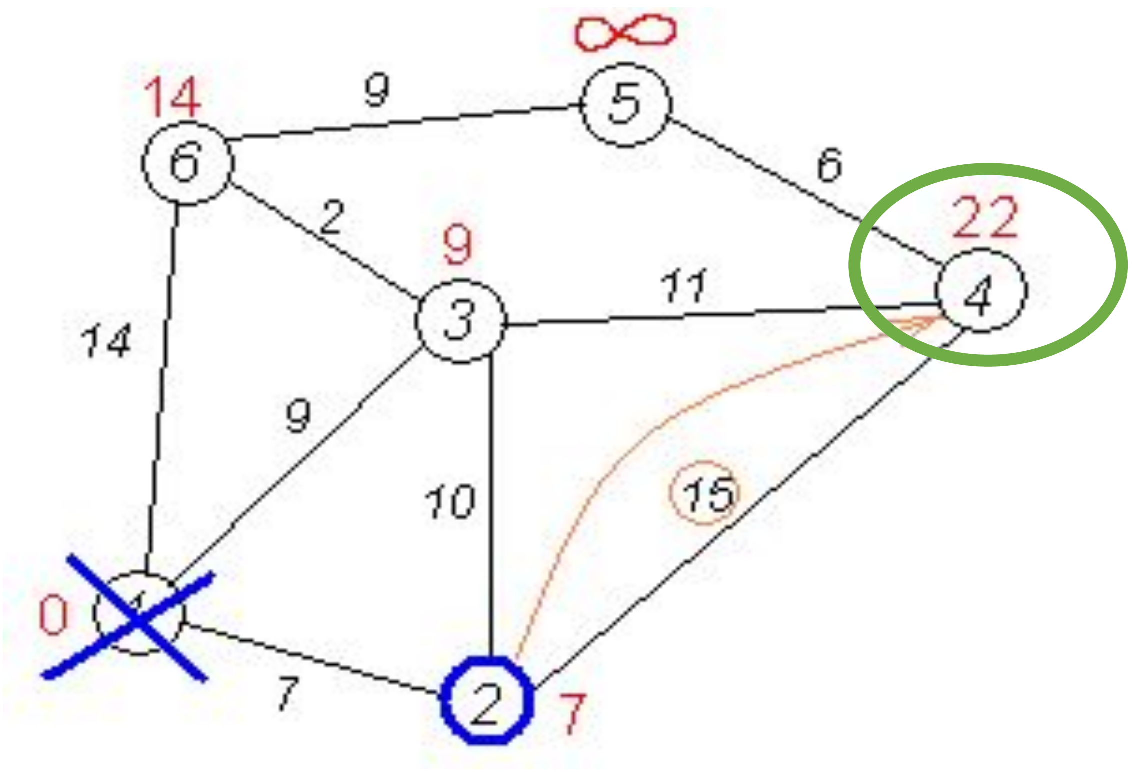
.



Крок 8 (3-4). Пошук шляхів до сусідів

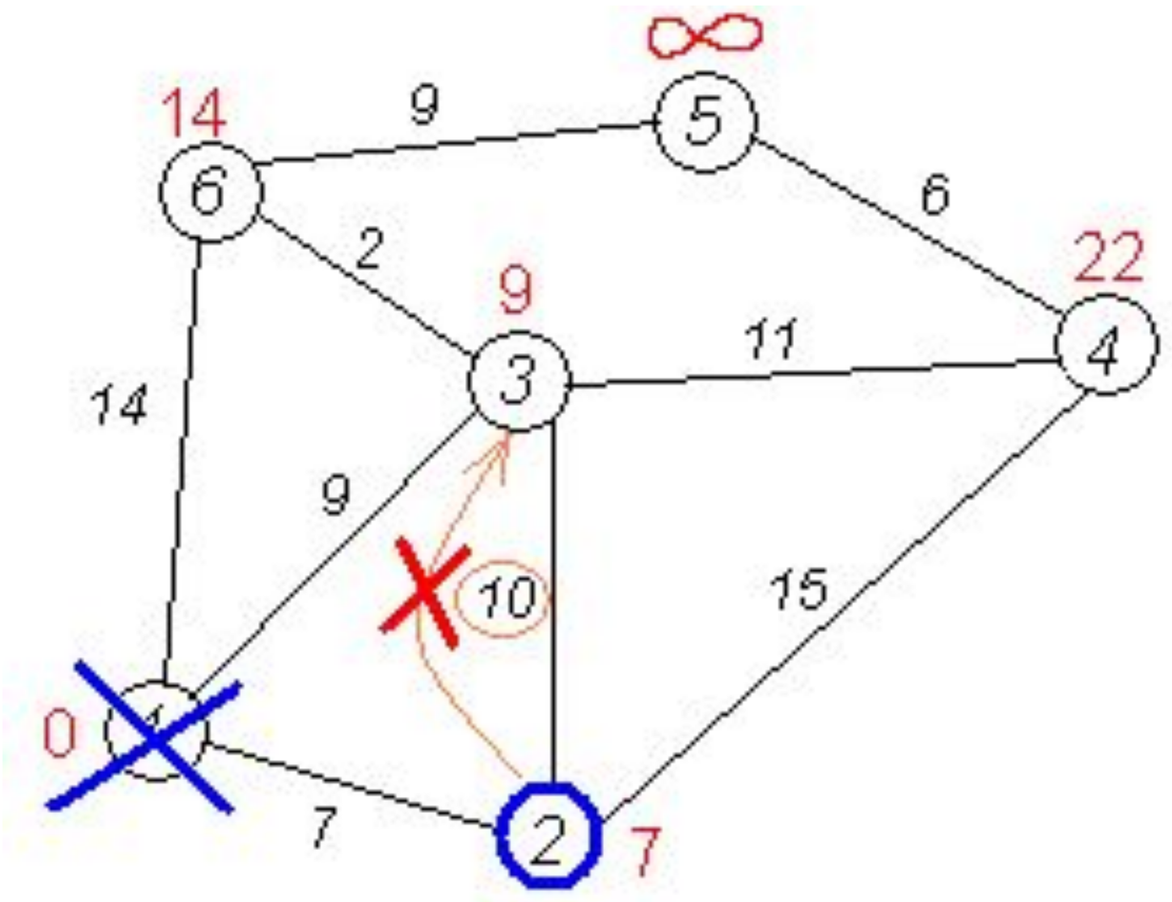
Перший сусід вершини № 2 — 1-ша вершина. Але вона обраблена (викреслена на кроці 6). Тому з 1-ю вершиною нічого не робимо.

Інший сусід вершини 2 — вершина 4. Якщо йти в неї через 2-у, то шлях буде = найкоротша відстань до 2-ї + відстань між 2-ю і 4-ю вершинами = 7 + 15 = 22. Оскільки 22 < ∞, встановлюємо відстань до вершини № 4 рівним 22 (червона стрілка).



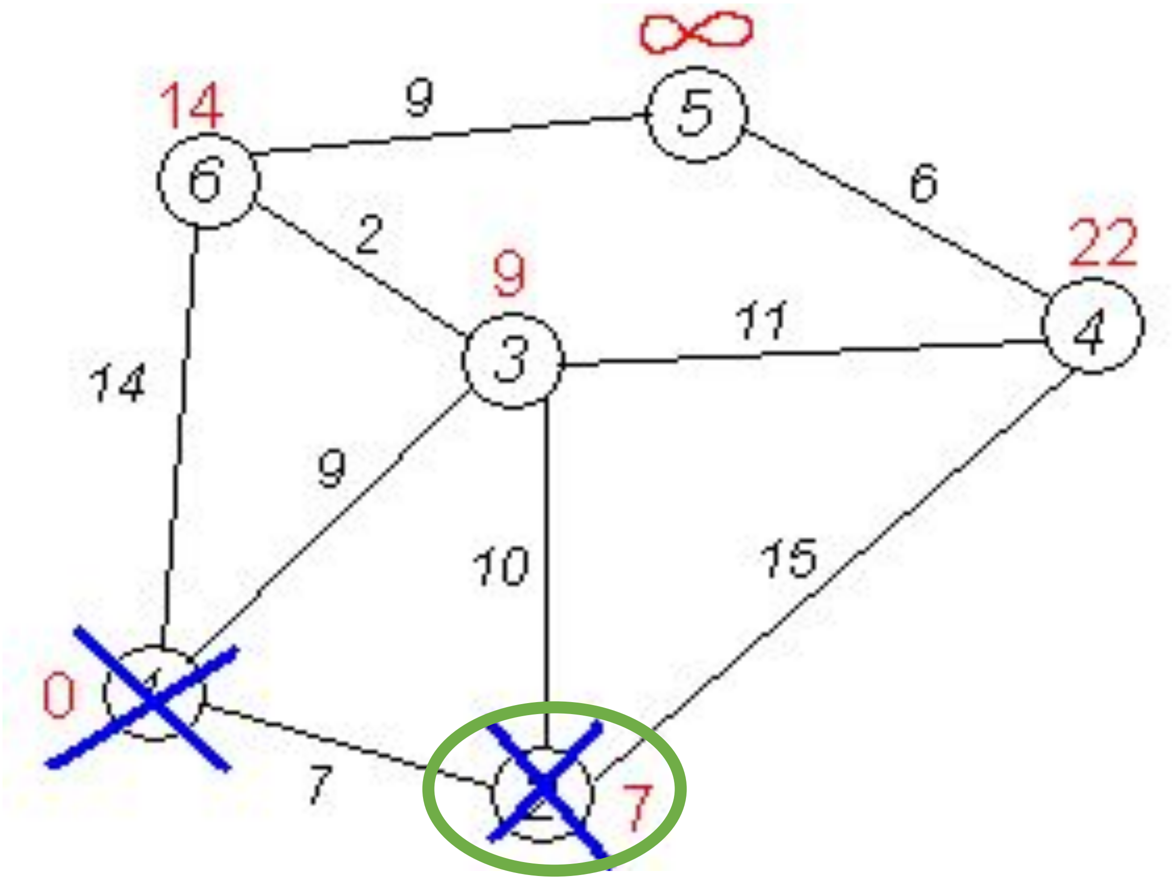
Крок 9 (5). Пошук шляху до останнього (з трьох) сусіда

Ще один сусід вершини 2 — вершина 3. Якщо йти в неї через 2-у, то шлях буде = 7 + 10 = 17. Але 17 більше за відстань, що вже запам'ятали раніше до вершини № 3 і дорівнює 9, тому поточну відстань до 3-ї вершини не міняємо (закреслена червона стрілка до неї).



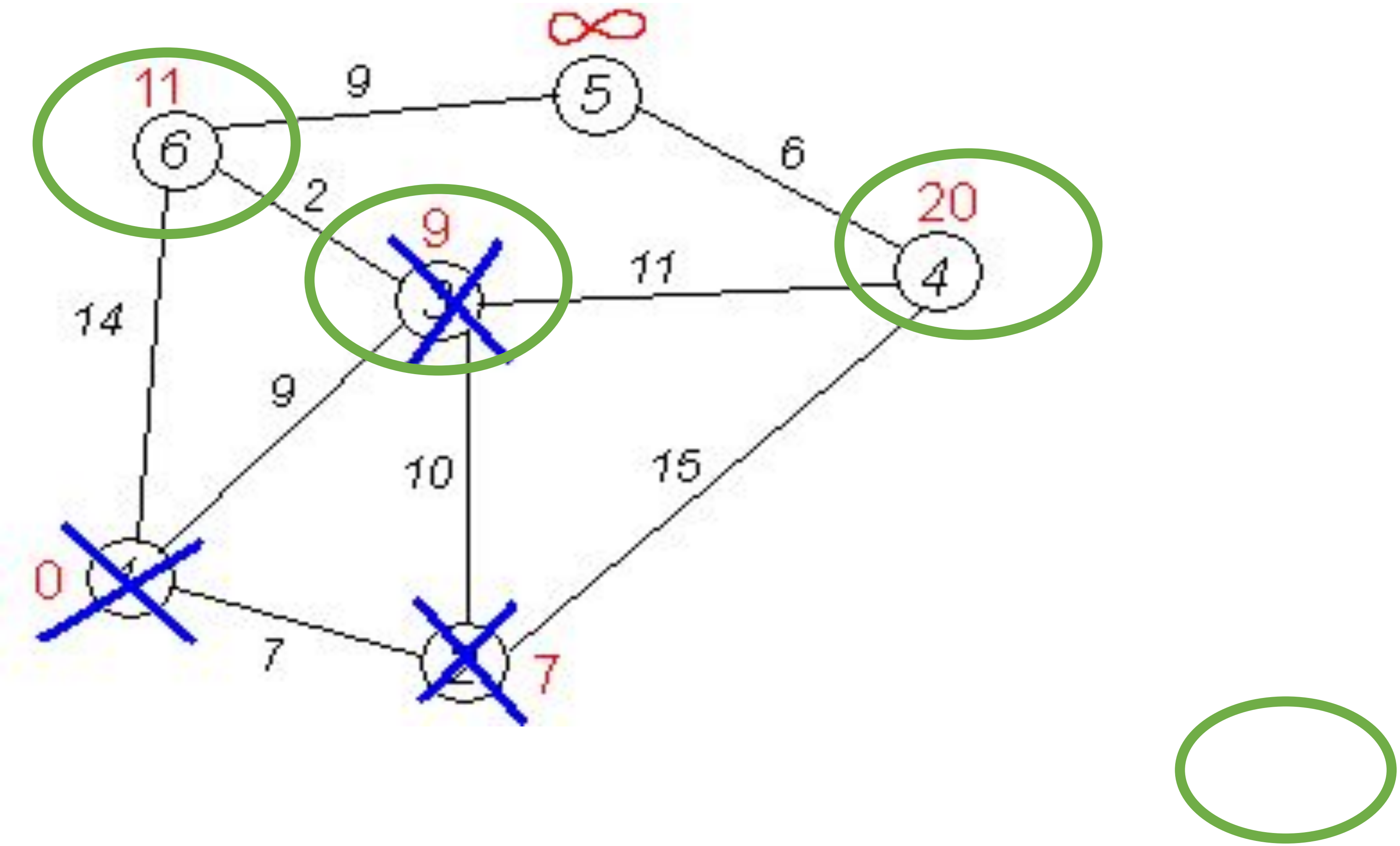
Крок 10 (6). Кінець опрацювання вершини 2.

Всі сусіди вершини 2 переглянуті, заморожуємо відстань до неї і викреслюємо її з графу.



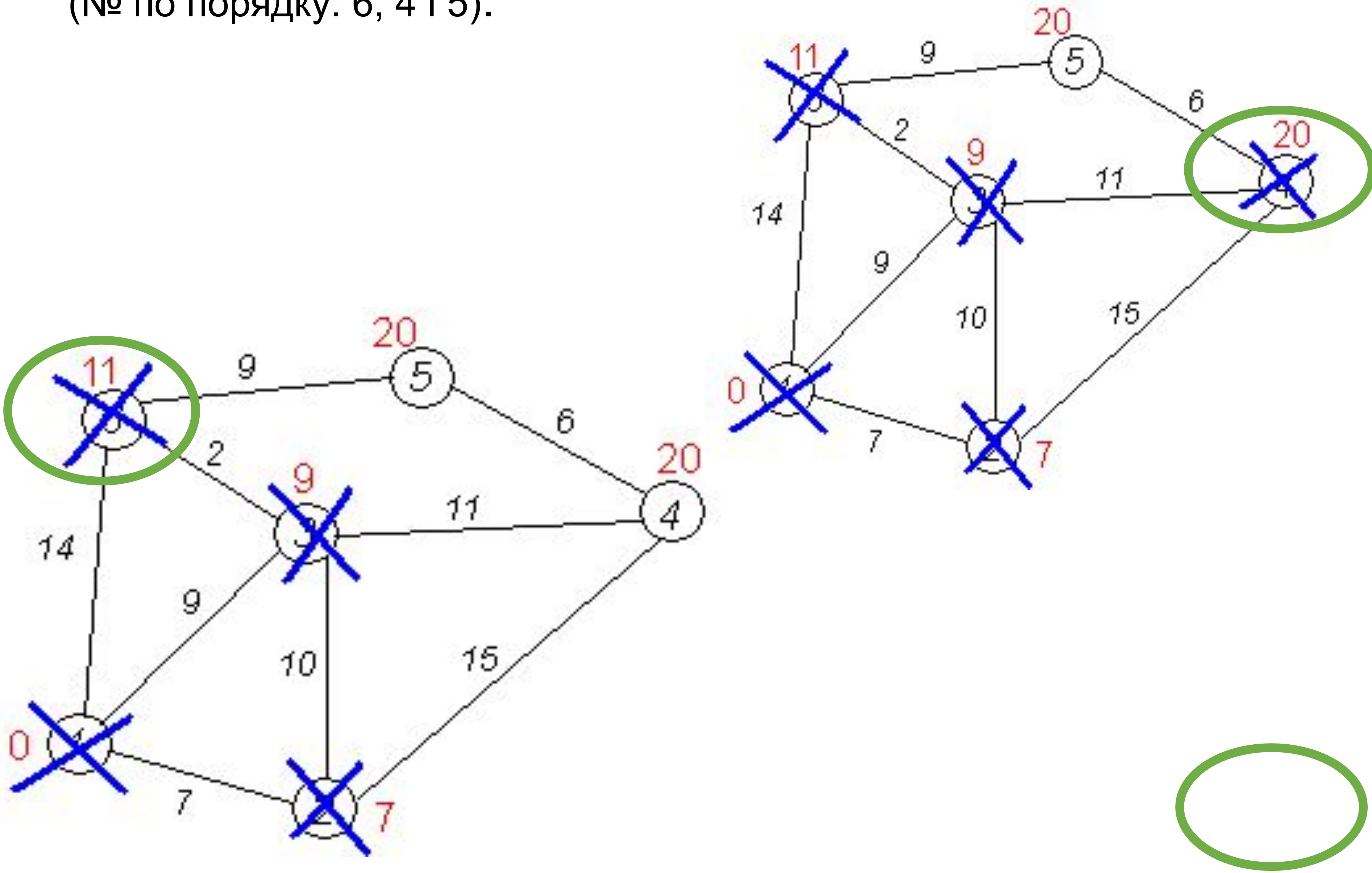
Кроки 11-15 (2-6). Процесс опрацювання вершини 3.

Всі сусіди вершини 2 переглянуті, заморожуємо відстань до неї і викреслюємо її з графу..



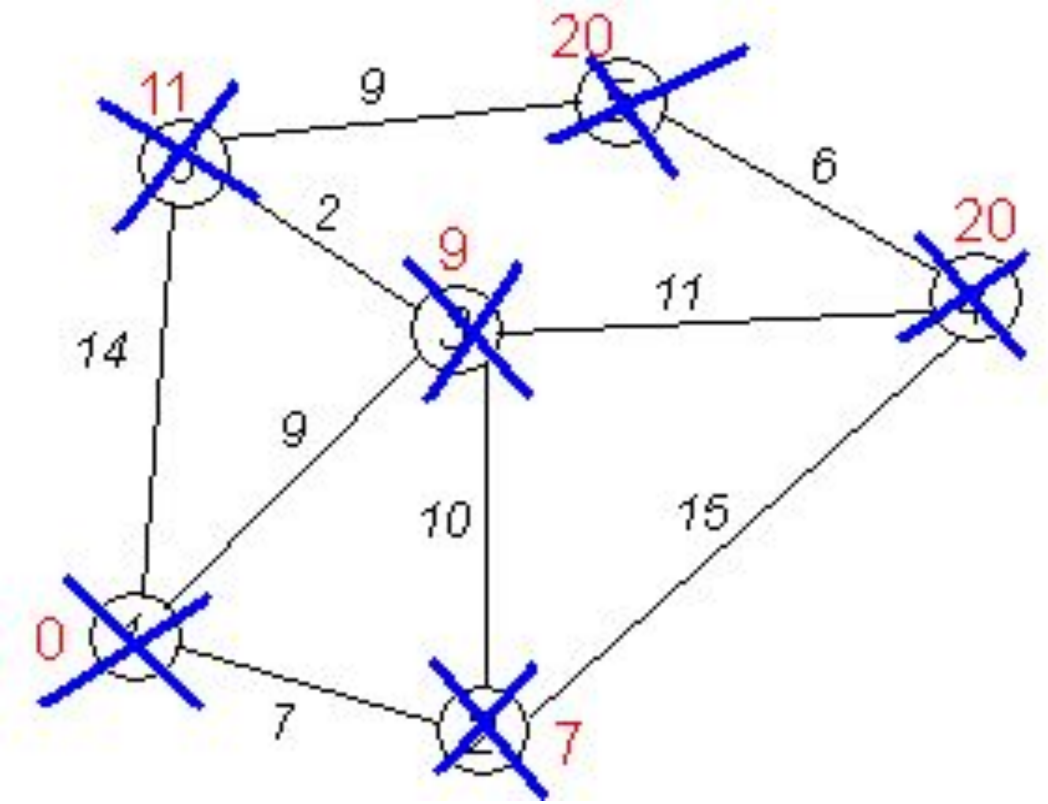
Наступні кроки. Опрацювання вершин 6, 4, 5.

Пророблюємо те саме з вершинами, що залишилися (№ по порядку: 6, 4 і 5).



Завершення виконання алгоритму.

Проробляємо те саме з вершинами, що залишилися (№ по порядку: 6, 4 і 5).



Алгоритм закінчує роботу, коли викреслені всі вершини. Результат його роботи видно на останньому рисунку: найкоротший шлях від 1-ї вершини до 2-ї становить 7, до 3-ї — 9, до 4-ї — 20, до 5-ї — 20, до 6-ї — 11 умовних одиниць.