

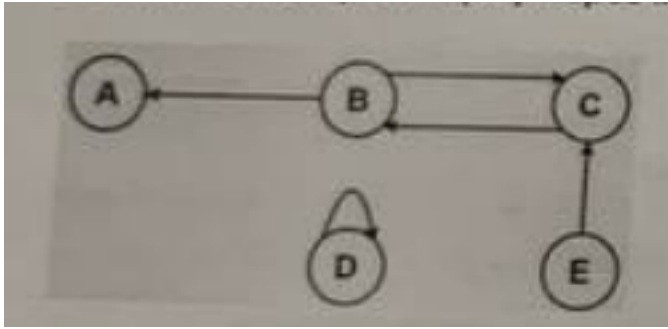


Частина 1

- Подати граф через матрицю суміжності:



	A	B	C	D	E
A	0	1	0	1	0
B	1	0	0	1	0
C	0	0	0	1	0
D	1	1	1	0	1
E	0	0	0	1	0



(ДЛЯ ОРГРАФА)

	A	B	C	D	E
A	0	0	0	0	0
B	1	0	1	0	0
C	0	1	0	0	0
D	0	0	0	1	0
E	0	0	1	0	0

- Нехай для заданого графа здійснено зображений пошук в глибину (біля вершин - мітки часу).
Класифікувати вказані ребра: ребро дерева, пряме, зворотнє, перехресне

🔍 Типи ребер в орграфі (за DFS та мітками часу):

Під час обходу графа в глибину всі ребра можна класифікувати на такі типи:

1. Ребро дерева (Tree Edge)

- Ребро, яким DFS переходить у ще не відвідану вершину.
- Входить до дерева обходу.
- Формально: якщо вершина v — ще біла (не відвідана) на момент обробки ребра $u \rightarrow v$.

2. Зворотнє ребро (Back Edge)

- Ребро до предка в дереві DFS (у тому ж ланцюжку рекурсії).
- Вказує на цикл в орграфі.
- Формально: вершина v — сіра (тобто обробка ще не завершена), коли обробляється $u \rightarrow v$.
- За мітками часу:

$$tin[v] < tin[u] < tout[u] < tout[v]$$

3. Пряме ребро (Forward Edge)

- Ребро з вершини u до нащадка v , який уже повністю оброблений.
- Не входить до дерева DFS, але веде "вниз".
- Формально: вершина v — чорна, і є нащадком u .
- За мітками часу:

$$tin[u] < tin[v] < tout[v] < tout[u]$$

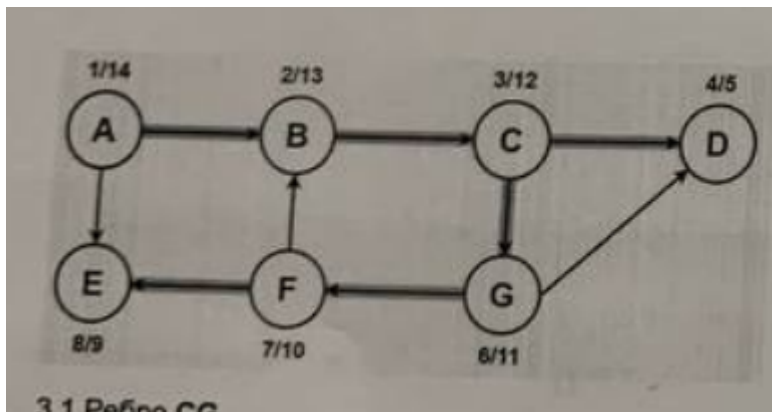
4. Перехресне ребро (Cross Edge)

- Ребро між різними гілками DFS, або з правої гілки в ліву.
- Не з'єднує предка й нащадка.
- Формально: вершина v — чорна, і не є нащадком u .
- За мітками часу:

$$tout[v] < tin[u] \quad \text{або} \quad tout[u] < tin[v]$$

🔄 Колір вершин у DFS:

- Біла (white) — ще не відвідана
- Сіра (gray) — відкрита, але ще не завершено обробку
- Чорна (black) — оброблена повністю



1. Ребра: **CG, FB**

CG: ребро дерева (T)

FB: зворотнє (B)

2. Ребра **AE, GD**

AE: пряме(F)

GD: перехресне (C)

- Для заданого орграфа перелічити його компоненти сильної зв'язності:

Лек 11 с 48

💡 Суперпроста шпаргалка: Як знайти КСЗ

1. Шукай цикли

Якщо кілька вершин утворюють замкнутий шлях (можна пройти туди-назад між усіма) — це одна компонента.

🔄 Наприклад:

$A \rightarrow B \rightarrow C \rightarrow A \rightarrow$ ✓ Компонента $\{A, B, C\}$

2. Вершина з одностороннім зв'язком

Якщо вершина має тільки вхід або тільки вихід — вона сама по собі.

➡ Наприклад:

$A \rightarrow B \rightarrow C$, але немає зворотного шляху

→ Кожна вершина — окрема КСЗ: $\{A\}$, $\{B\}$, $\{C\}$

3. Шукай "замикаючі" ребра назад

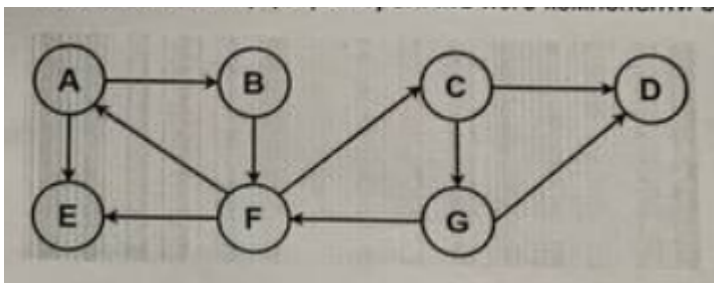
Якщо якась вершина замикає цикл назад — вся петля входить в одну компоненту.

🔄 Наприклад:

$G \rightarrow F, F \rightarrow A \rightarrow B \rightarrow G$

→ G може повернутись до себе через цей цикл

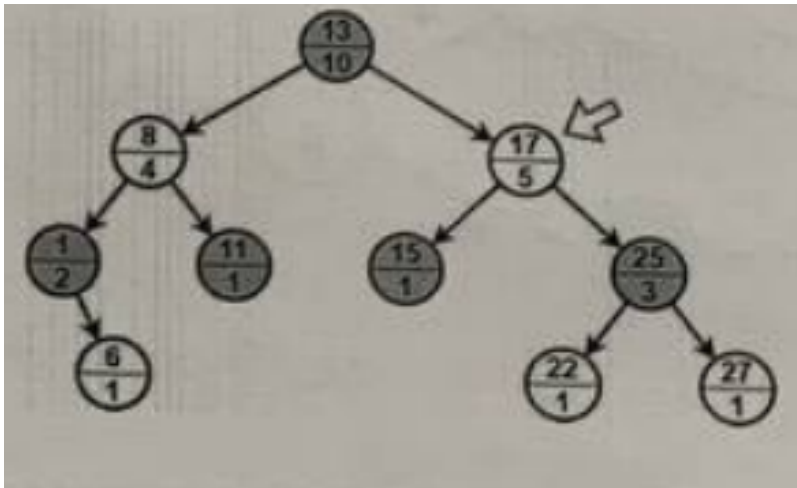
✓ Тоді $\{A, B, F, G\}$ — одна сильнозв'язна компонента



Компоненти сильної зв'язності: $\{ABFCG\}$ $\{D\}$ $\{E\}$

Чатсина 2

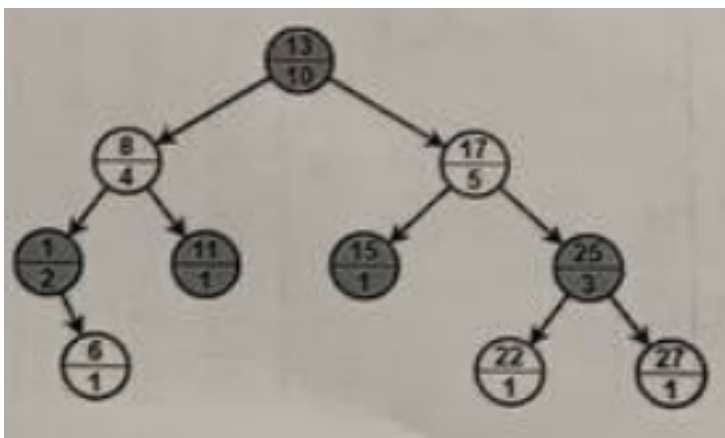
- Для поданого дерева порядкової статистики визначити ранг вказаного елемента:



(По факту якщо впорядкуємо ключі дерева у масив за зростанням то позиція шуканого елемента й буде його рангом: [1,6,8,11,13,15,17,22,25,27])

Ранг вузла 17/5: 7

- Для поданого дерева порядкової статистики вкажіть значення ключа елемента з рангом 5

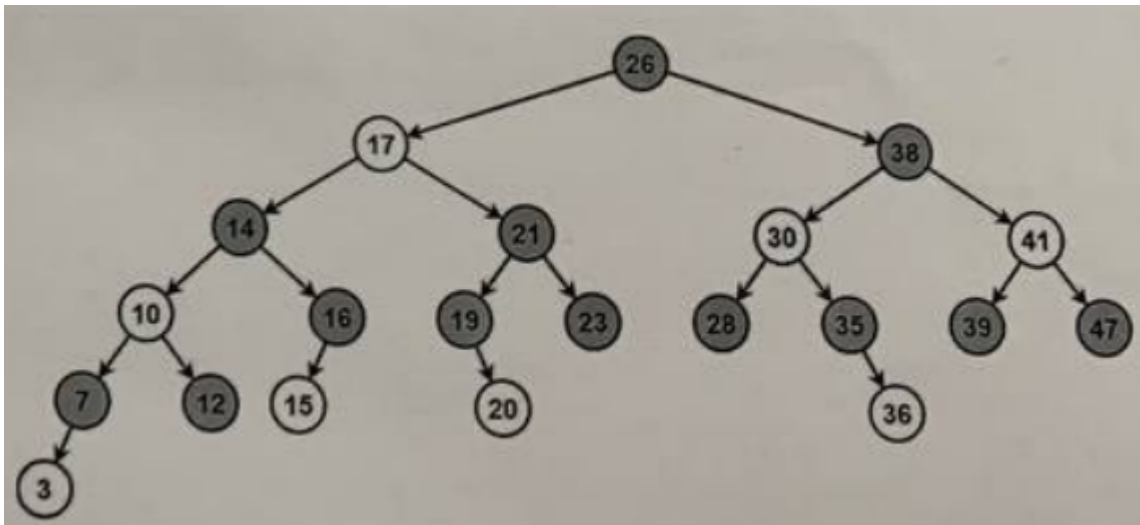


[1,6,8,11,13,15,17,22,25,27]

Елемент з рангом 5: 13

- Якою є чорна висота вузлів з наступними ключами зображеного червоно-чорного дерева

- Чорна висота вузла x ($bh(x)$, black-height) – кількість чорних вузлів на шляху від вузла x (не рахуючи його самого) до листів; вона визначається однозначно.
- Чорна висота дерева – чорна висота його кореня.



1.

Ключ 14: 2

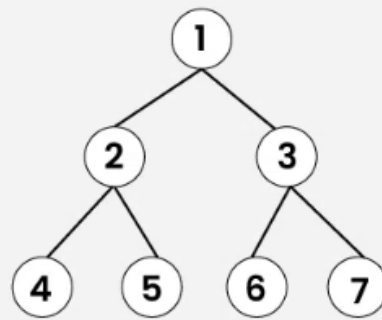
Ключ 36: 1

2.

Ключ 10: 2

Ключ 38: 2

Tree Traversal Techniques

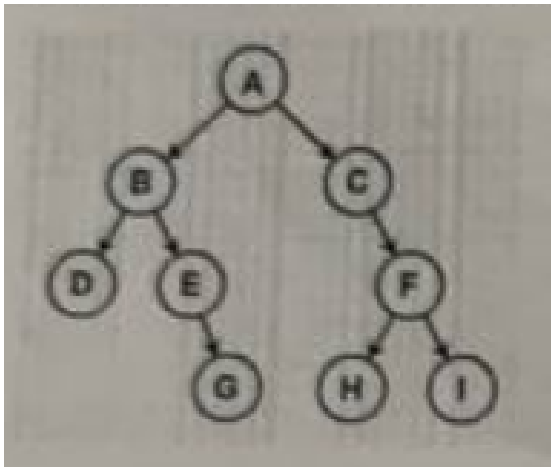


Inorder Traversal						
4	2	5	1	6	3	7

Preorder Traversal						
1	2	4	5	3	6	7

Postorder Traversal						
4	5	2	6	7	3	1

Level order Traversal						
1	2	3	4	5	6	7



- Вкажіть послідовність вершини при прямому порядку обходу дерева (PREORDER)

[A,B,D,E,G,C,F,H,I]

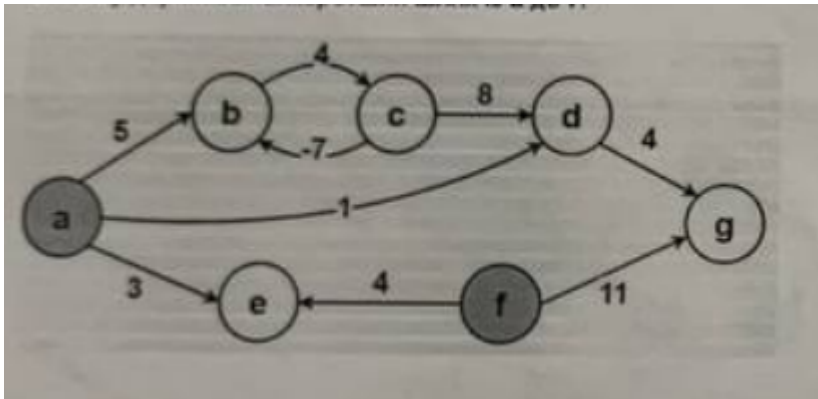
- Вкажіть послідовність вершини при симетричному порядку обходу дерева(INORDER)

[D,B,E,G,A,C,H,F,I]

- !!! Чому дорівнює найкоротший шлях

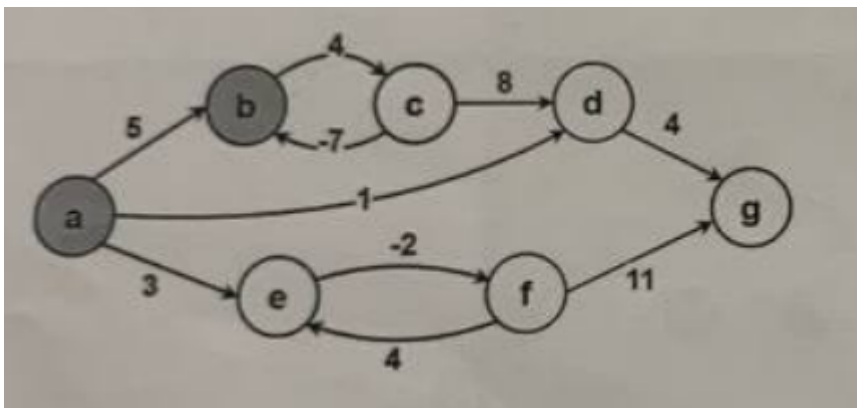
(Я не впевнений за якими алгоритмами тут правильно шукати)

1. Із $a \rightarrow f$



inf

2. Із $a \rightarrow b$



-inf

- В яку комірку хеш таблиці потрапить значення за умови лінійного дослідження та допоміжної хеш функції $h'(K) = K$

9. В яку комірку хеш-таблиці потрапить значення 35 за умови лінійного дослідження та допоміжної хеш-функції $h'(K) = K$?

0	1	2	3	4	5	6	7	8	9	10
	23	79	3			61	17			65

$$h(k) = k \bmod 11$$

$35 \bmod 11 = 2$ комірка 2 зайнята тому 35 буде у комірці 4 ($i=2$)

9. В яку комірку хеш-таблиці потрапить значення 25 за умови лінійного дослідження та допоміжної хеш-функції $h(K) = K$?

0	1	2	3	4	5	6	7	8	9	10
	12		36	48		28	18	41		

$$h(k) = k \bmod 11$$

$25 \bmod 11 = 3$ комірка 3 зайнята тому 35 буде у комірці 5 ($i=2$)

Чатсина 3

10. Які з дерев пошуку не містять додаткових полів з інформацією для балансування? GPT

- a) червоно-чорне дерево
- b) AVL-дерево
- c) splay-дерево
- d) B-дерево

11. Якою може бути максимальна кількість обертань при модифікації AVL-дерева? GPT

- a) 1
- b) 2
- c) 3
- d) 4
- e) залежить від висоти дерева

12. Що з вказаного вірно для B-дерева? GPT

- a) всі вузли містять однакову кількість ключів
- b) значення ключів у вузлі впорядковані у неспадному порядку
- c) всі неплисткові вузли мають однакову кількість потомків
- d) всі значення ключів вузла не менші за ключі вузла-потомків