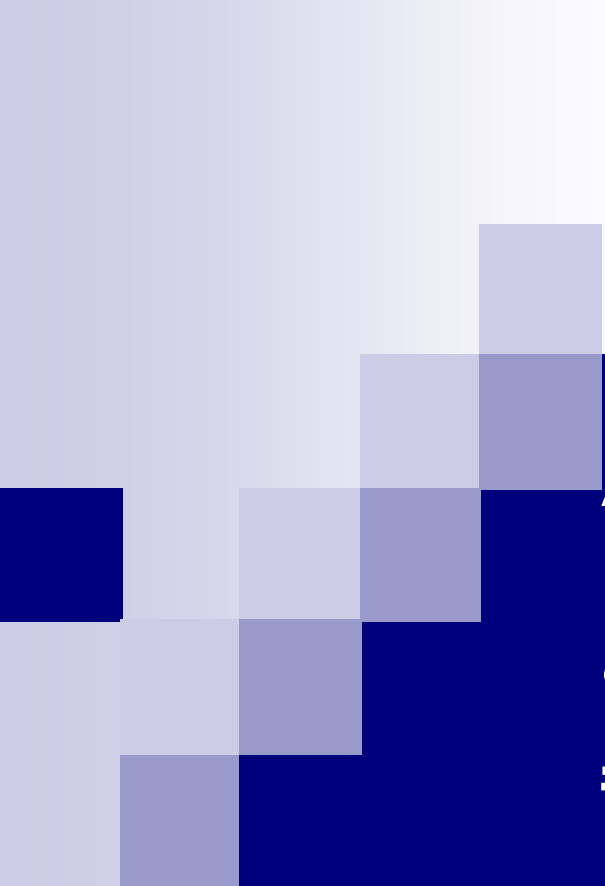




Алгоритми та структури даних

Модуль 2: Методи розробки алгоритмів

Лекція 10. Метод перетворення.

Кудін Олексій Володимирович
avk256@gmail.com



План

1. Визначення
2. Попереднє сортування.
3. Збалансовані дерева пошуку.
4. Визначення піраміди.
5. Пірамідальне сортування (самостійно).

- *Яка основна ідея метода декомпозиції?*
- *Яка часова складність сортування злиттям та швидкого сортування?*
- *Що таке бінарне дерево пошуку?*

- Перетворення в більш простий або більш зручний для рішення екземпляр тої ж задачі - спрощення екземпляра.
- Зміна подання наявного екземпляра задачі.
- Приведення задачі, тобто перетворення до екземпляра іншої задачі, для якої є алгоритм рішення.

Приклад 1. Перевірка унікальності елементів масиву.

АЛГОРИТМ *PresortElementUniqueness* ($A[0..n - 1]$)

// Проверка единственности элементов массива

// **Входные данные:** Массив $A[0..n - 1]$ упорядочиваемых элементов

// **Выходные данные:** **true**, если в A нет одинаковых элементов,

// **и false**, если есть

Сортировка массива A

for $i \leftarrow 0$ **to** $n - 2$ **do**

if $A[i] = A[i + 1]$

return false

return true

Приклад 2. Обчислення моди.

АЛГОРИТМ *PresortMode* ($A[0..n-1]$)

// Вычисляет моду массива с использованием

// предварительной сортировки

// **Входные данные:** Массив $A[0..n-1]$ упорядочиваемых элементов

// **Выходные данные:** Мода массива

Сортировка массива A

$i \leftarrow 0$ // Текущее сканирование начинается с позиции i

$modefrequency \leftarrow 0$ // Максимальное количество одинаковых элементов

while $i \leq n-1$ **do**

$runlength \leftarrow 1$; $runvalue \leftarrow A[i]$

while $i + runlength \leq n-1$ **and** $A[i + runlength] = runvalue$ **do**

$runlength = runlength + 1$

if $runlength > modefrequency$

$modefrequency \leftarrow runlength$; $modevalue \leftarrow runvalue$

$i \leftarrow i + runlength$

return $modevalue$

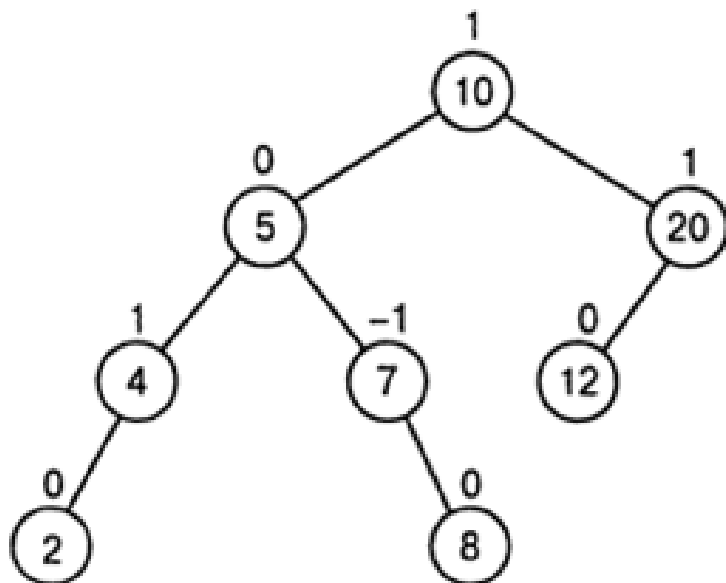
Приклад 3. Обчислення перетину двох множин (самостійно).

Бінарне дерево пошуку — це бінарне дерево, вузли якого містять елементи множини елементів, що впорядковуються, по одному елементу у вузлі, причому всі елементи у лівому піддереві менше елемента в корені піддерева, а елементи в правому піддереві - більше його.

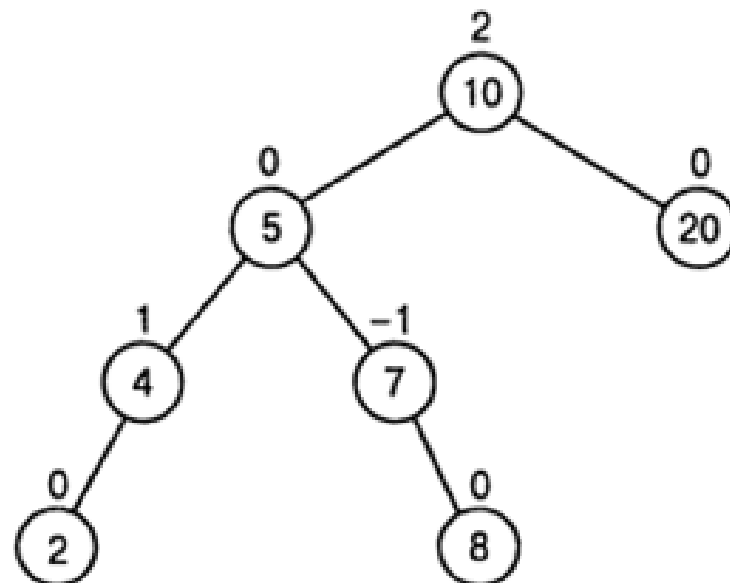
Проблема — вироджене дерево, коли всі елементи списку записано в одні гілці бінарного дерева.

1. AVL-дерева.
2. 2-3 дерева.

AVL-деревя – це бінарне деревя пошуку, у якому показник збалансованості (balance factor) кожного вузла, що визначається як різниця висот лівого й правого піддерев вузла, дорівнює 0, +1 або -1.

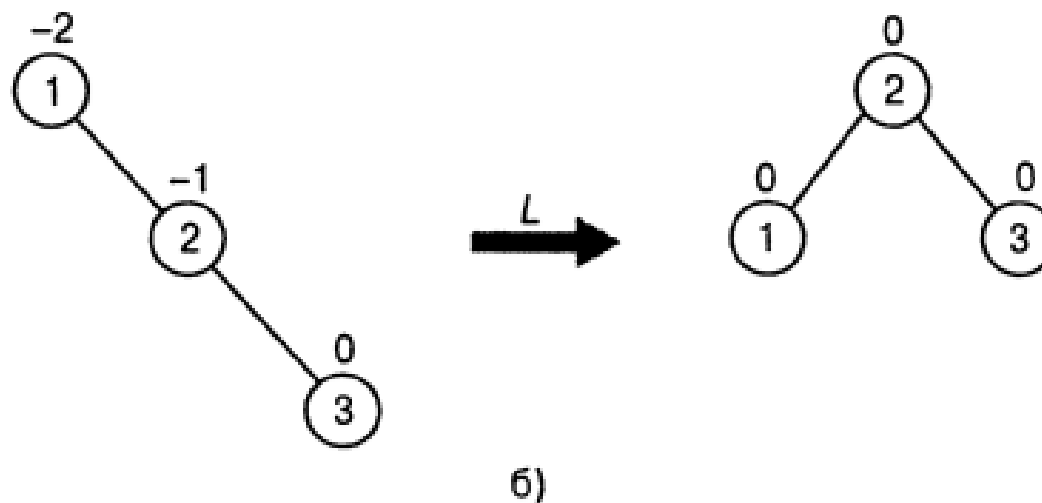
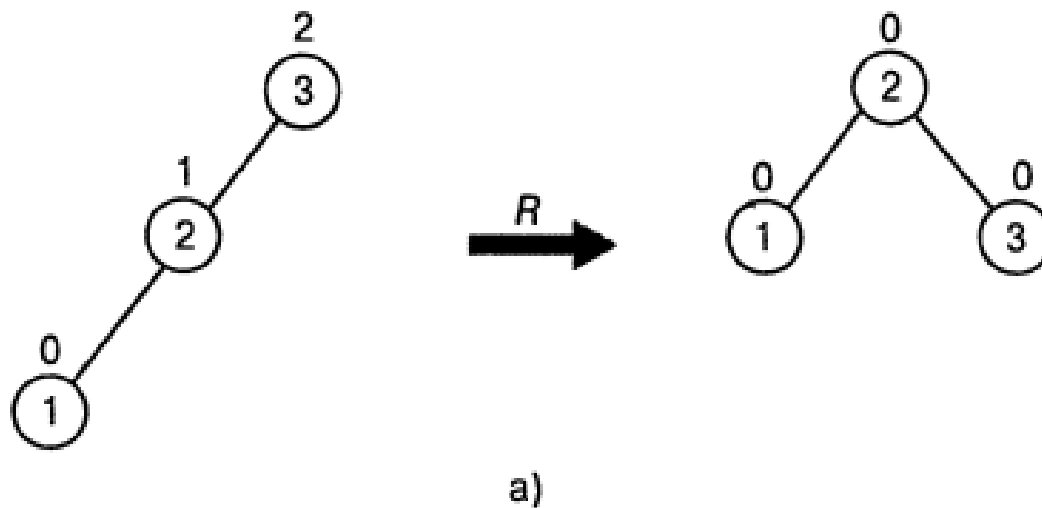


а)

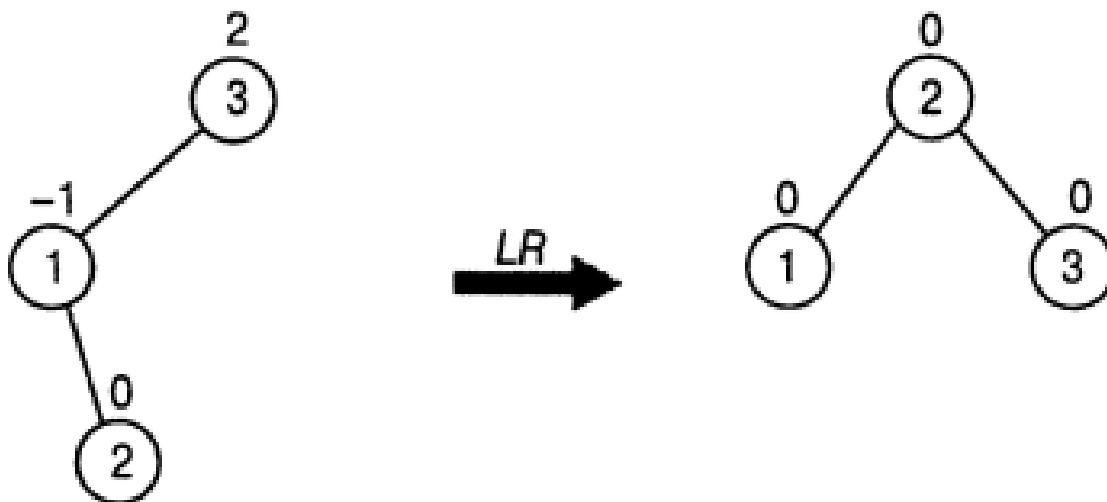


б)

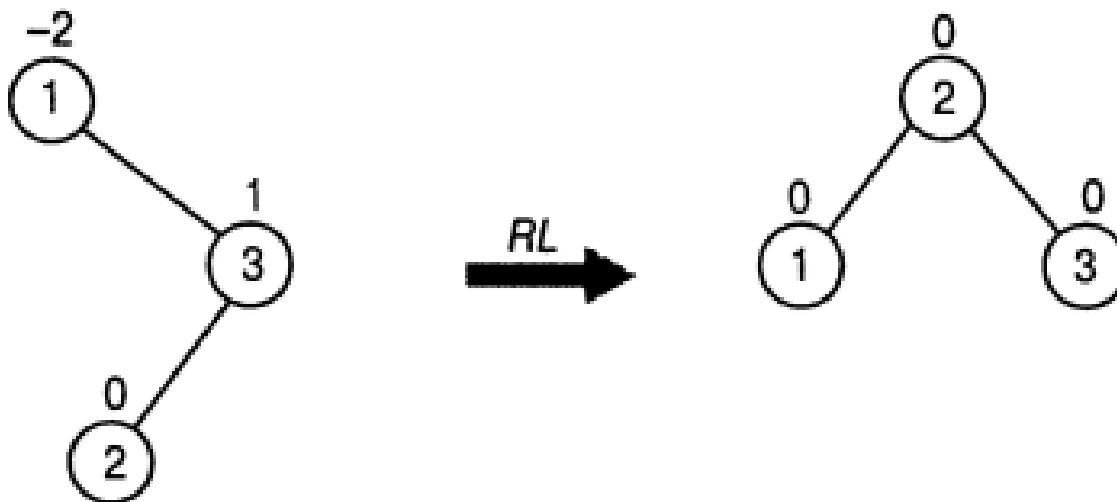
Перетворення не AVL-деревя:



Комбіновані повороти AVL-деревя:

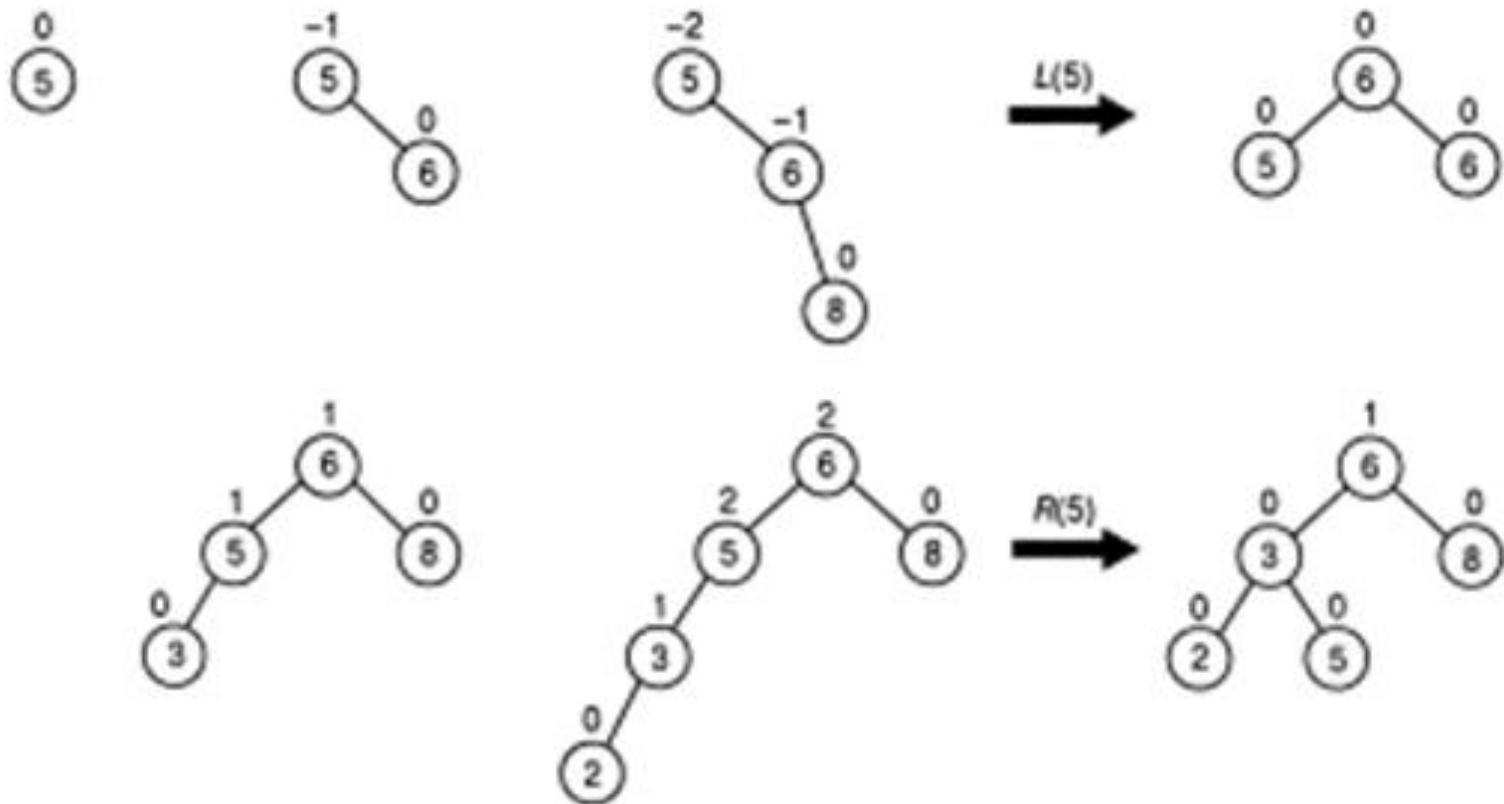


в)



г)

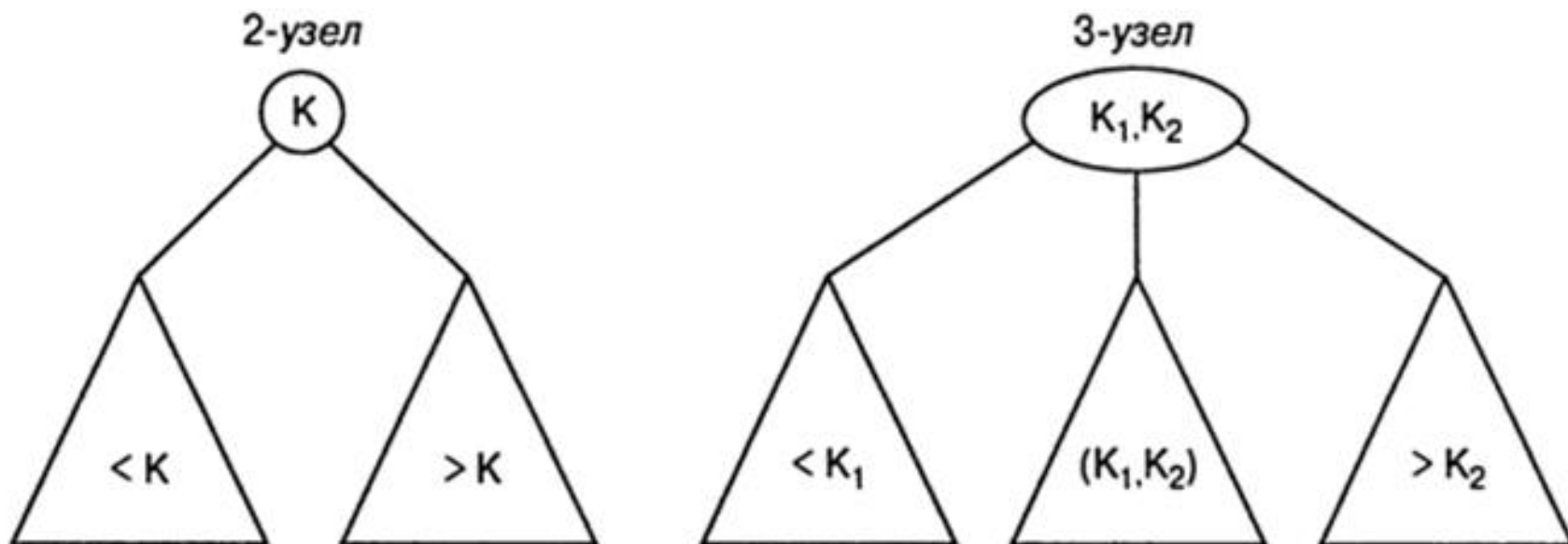
Приклад перетворення до AVL-дерев
послідовності 5, 6, 8, 3, 2:



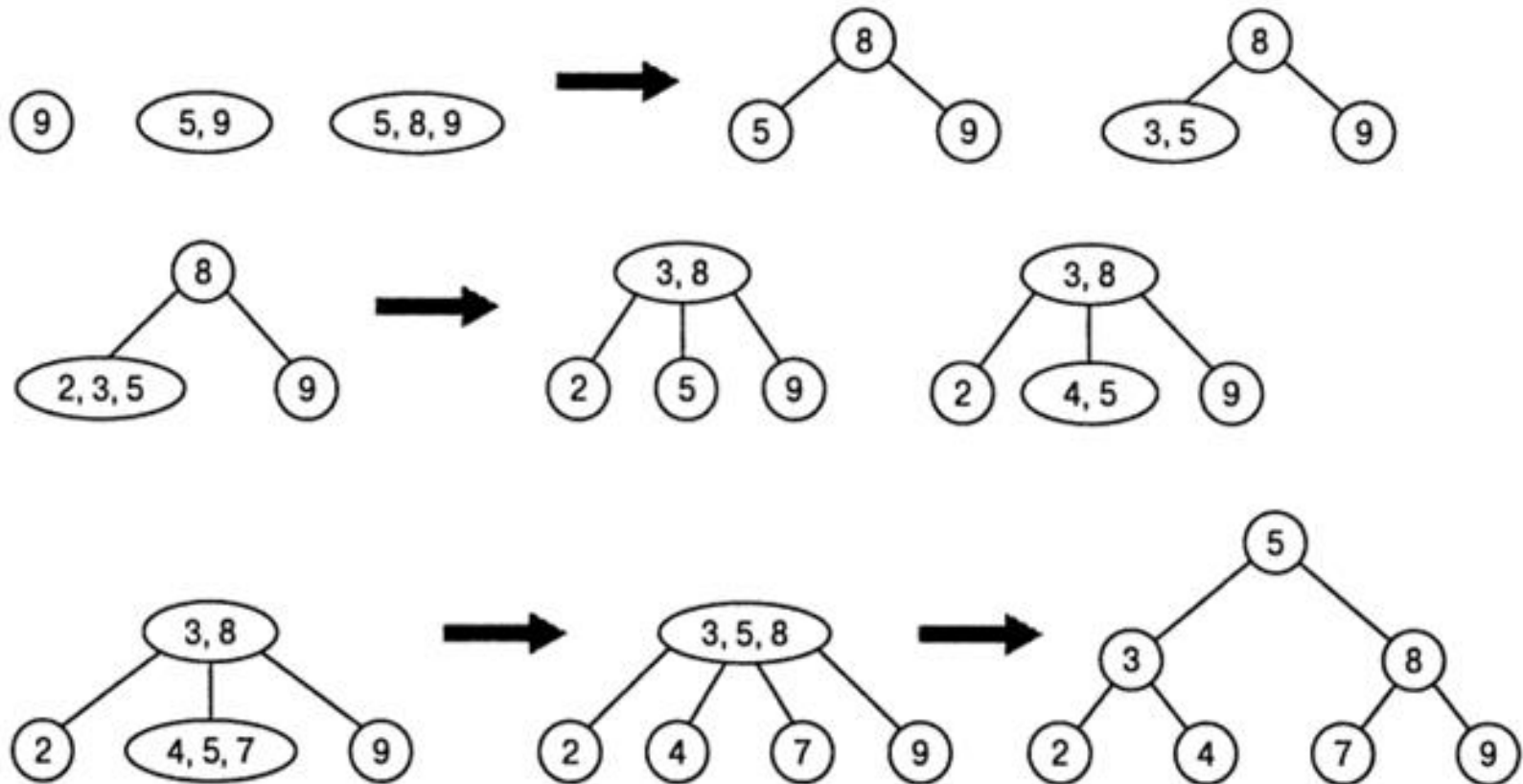
2-3-дерево – це дерево пошуку, що може мати вузли двох видів — 2-вузли й 3-вузли. **2-вузол** містить єдиний ключ K і має два нащадка.

3-вузол містить два впорядкованих ключі K_1 і K_2 ($K_1 < K_2$) і має три дочірніх вузли.

Особливість: 2-3-дерево завжди збалансовано по висоті.



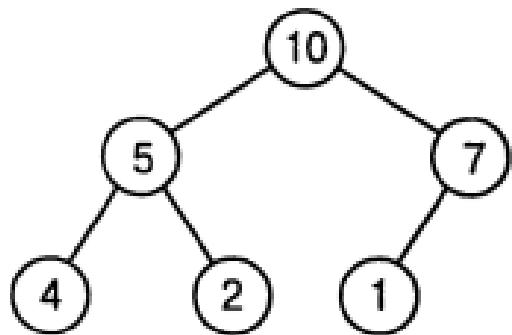
Приклад. Побудова 2-3 дерева для послідовності 9, 5, 8, 3, 2, 4, 7.



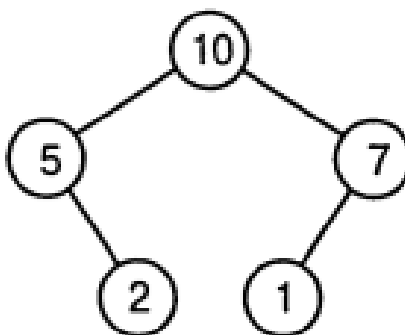
Піраміда (heap) може бути визначена як бінарне дерево, для якого виконуються дві наступні умови:

- **Вимога до форми дерева.** Всі рівні бінарного дерева заповнені, за винятком, можливо, останнього рівня, у якому можуть бути відсутніми деякі крайні праворуч листи.
- **Вимога домінування батьківських вузлів.** Ключ у кожному вузлі не менше ключів у його дочірніх вузлах.

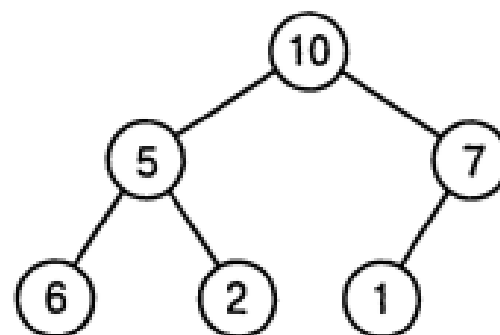
Яке з неведених дерев є пірамідою?



а)

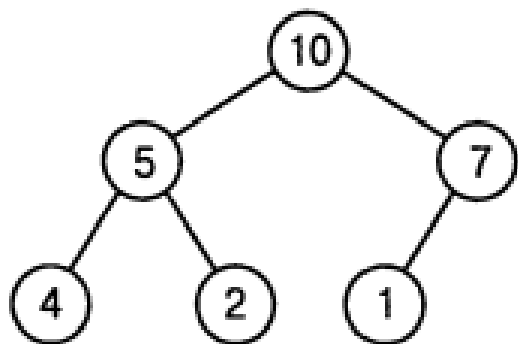


б)



в)

Представлення піраміди в програмі:



Представление в виде массива

Индекс	0	1	2	3	4	5	6
Значение		10	5	7	4	2	1
		Родительские узлы			Листья		

Рис. 6.10. Піраміда и ее представление в виде массива

Таким чином, ми можемо визначити піраміду як масив $H[1..n]$, у якому кожний елемент у позиції i у першій половині масиву більше або дорівнює елементів у позиціях $2i$ і $2i + 1$, тобто

$$H[i] \geq \max \{ H[2i], H[2i + 1] \} \quad \text{для } i = 1, 2, \dots, \lfloor n/2 \rfloor.$$

Резюме (що встигли за лекцію)

- Дали загальне визначення методу перетворення.
- Розглянули застосування попереднього сортування.
- Розглянули збалансовані дерева пошуку: AVL- та 2-3 дерева.
- Розглянули поняття піраміди.

До наступної лекції:

Розібрати алгоритм пірамідального сортування.