

# Купа (структура даних)

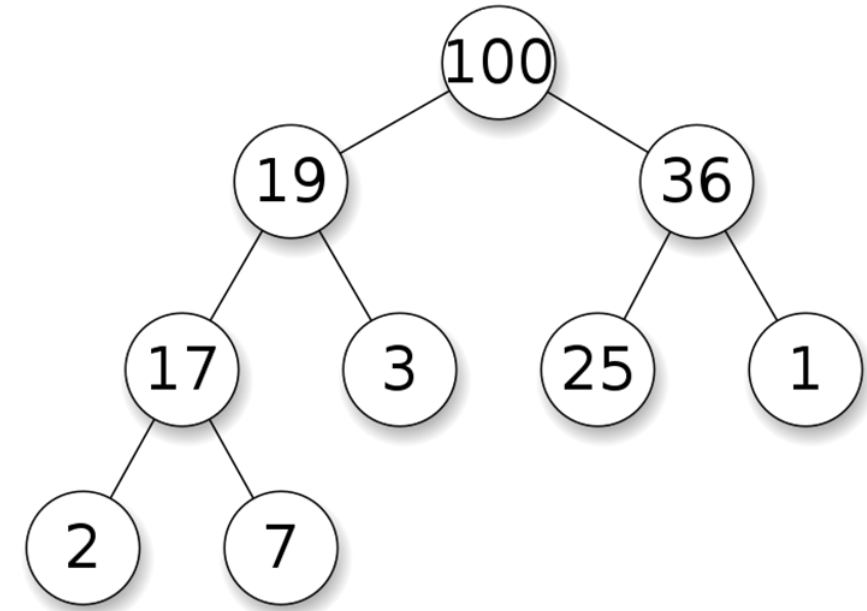
Прудієв Ігор

Гр. ФВЕ 1 к. маг.

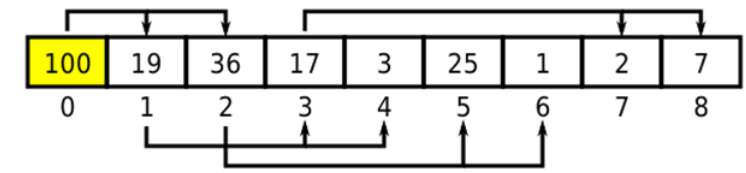
20.04.2021

# Означення

- Купа (heap) – спеціалізована деревовидна структура даних з певними властивостями впорядкованості:
- Якщо  $B$  – вузол-нащадок  $A$ , тоді  $\text{ключ}(A) \geq \text{ключ}(B)$  (так звані max-купи).
- Якщо нерівність перевернути, буде утворена min-купа.
- Не існує обмежень на кількість нащадків вузла, хоча зазвичай використовують не більше двох (бінарні купи). Для спрощення розглянемо саме бінарну.
- Висота бінарної купи  $h = \log_2 N$



## Array representation



# Складність операцій з купою

	Список	Список, сортований за зменшенням	Купа
Додавання елементу	$O(1)$	$O(n)$	$O(\log n)$
Пошук елементу	$O(n)$	$O(1)$	$O(\log n)$

Всі операції з купою потребують  $O(h)$  операцій. Нехай в купі  $n$  елементів, а її висота  $h$ . Тоді найбільше число елементів, яке може бути в купі висотою  $h$  є

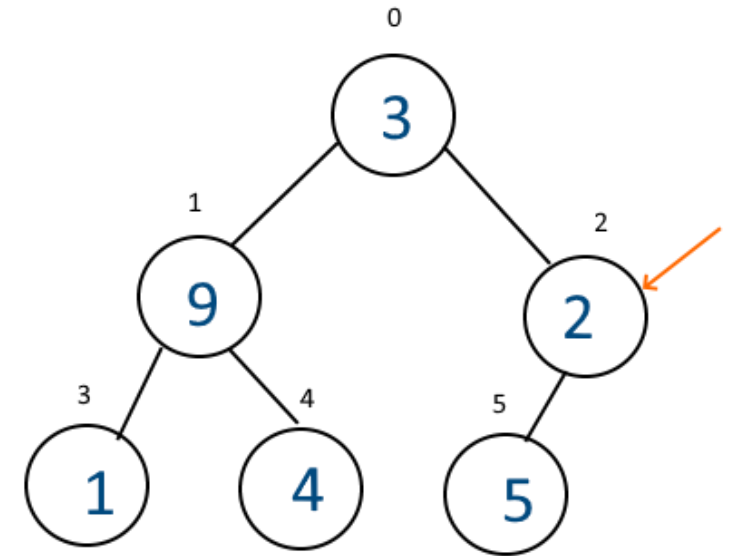
$$1 + 2 + 4 + \dots + 2^{h-1} = 2^h - 1.$$

Таким чином,  $2^{h-1} \leq n < 2^h$ , звідки видно, що  $h$  приблизно рівне логарифму числа  $n$ , тобто складність операцій з купою -  $O(\log n)$

# Операції над купою. Приведення до купи. (Heapify)

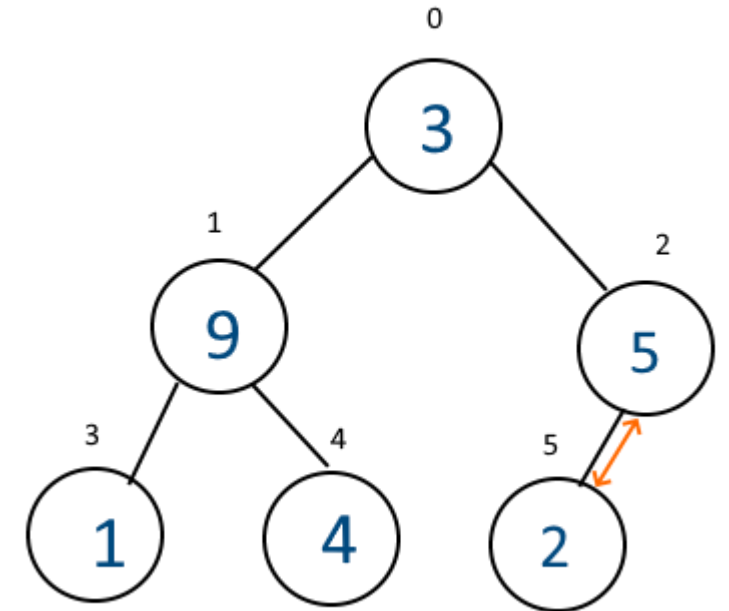
1. Візьмемо початковий масив
2. Утворюємо бінарне дерево
3. Починаємо з першого вузла, що не є листком, має індекс  $n/2 - 1$ .
4. Позначаємо цей елемент  $i$  як найбільший **largest**
5. Індекс лівого нащадка буде  $2i + 1$ , правого -  $2i + 2$ .

3 9 2 1 4 5  
0 1 2 3 4 5



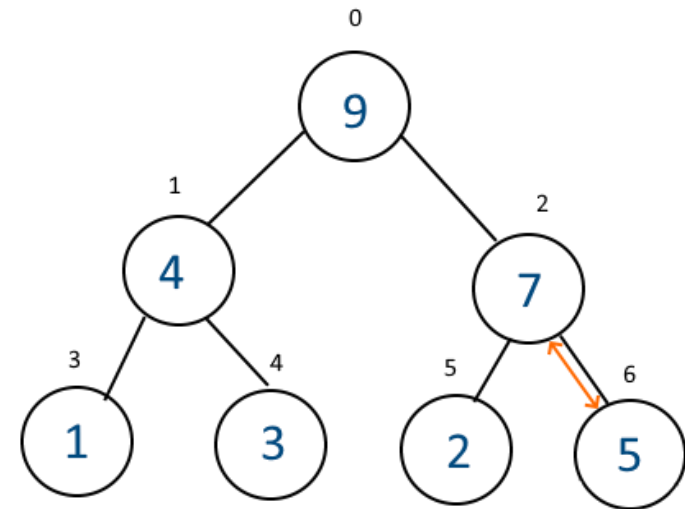
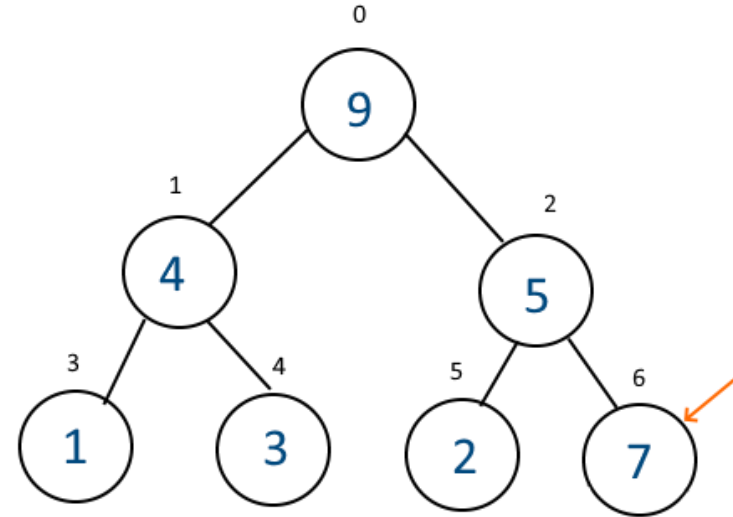
# Операції над купою. Приведення до купи. (Heapify)

- Якщо лівий нащадок **leftChild** більший за вибраний елемент **i**, вибираємо лівого нащадка як найбільший елемент **largest**.
  - Якщо правий нащадок **rightChild** більший за вибраний елемент **i**, вибираємо правого нащадка як найбільший елемент **largest**.
6. Міняємо місцями найбільший елемент з поточним.
  7. Повторюємо пункти 3-7 до поки піддерева приведуться до купи.
  8. Для утворення max-купи потрібно пройтися по індексам від першого не листочка до 0.



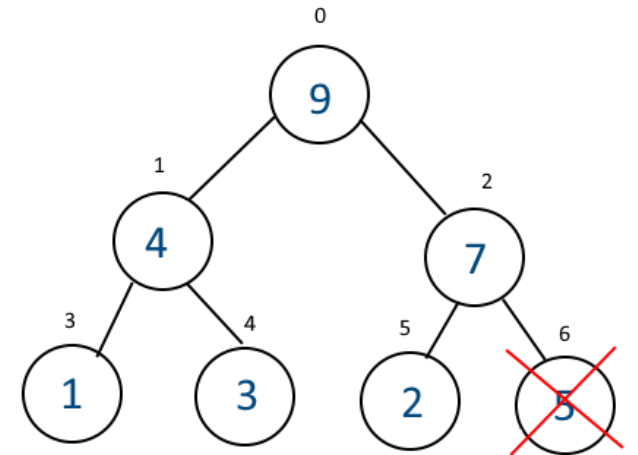
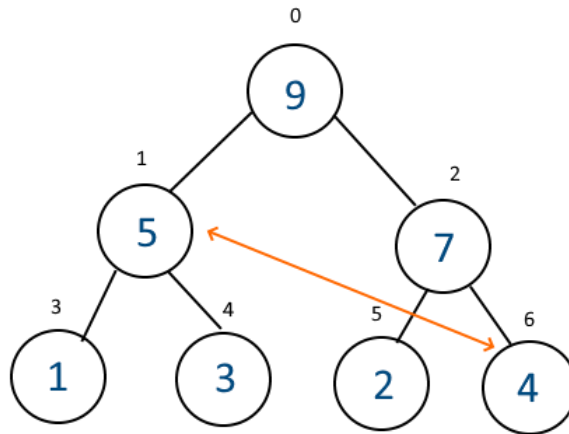
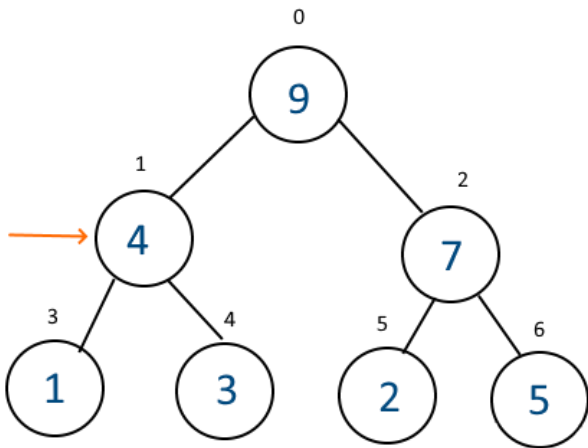
# Операції над купою. Додавання елементу.

1. Додаємо новий елемент в кінці дерева.
2. Виконуємо приведення до купи.



# Операції над купою. Видалення елемента.

1. Вибираємо елемент для видалення.
2. Міняємо його місцями з останнім елементом.
3. Видаляємо останній елемент.
4. Приводимо до купи.



# Приклади використання.

- Пірамідальне сортування. Не має квадратичних найгірших сценаріїв.  **$O(n \log n)$**
- Черга з пріоритетом.
- Алгоритми на графах. (як пошук найкоротшого шляху)