

## Лекція 26. Алгоритм сортування на дереві

Алгоритм пірамідального сортування (англійський термін Heap Sort) використовує представлення масиву довжини  $n$  у вигляді дерева з  $n$  вершинами. Оцінка швидкодії цього алгоритму дорівнює  $O(n \log n)$ . Цей алгоритм не вимагає допоміжних масивів, сортуючи “на місці”, тобто оцінка необхідного обсягу пам’яті  $O(1)$ . Розглянемо спочатку метод представлення масиву у виді дерева:

Нехай  $A[1 .. n]$  - деякий масив. Зіставимо йому дерево, використовуючи наступні правила:

1.  $A[1]$  - корінь дерева ;
2. Якщо  $A[i]$  - вузол дерева, то  $A[2*i]$  - вузол - “лівий син” вузла  $A[i]$
3. Якщо  $A[i]$  - вузол дерева, то  $A[2*i+1]$  - вузол - “правий син” вузла  $A[i]$

Правила 1-3 визначають у масиві структуру дерева, причому глибина дерева не перевершує  $\lceil \log_2 n \rceil + 1$ . Вони ж задають спосіб руху по дереву від кореня до листків. Рух вгору задається правилом 4:

4. Якщо  $A[i]$  - вузол дерева та  $i > 1$ , то  $A[i \text{ mod } 2]$  - вузол - “батько” вузла  $A[i]$ .

Для роботи з алгоритмом пірамідального сортування, нам необхідно ввести елементи масиву визначеної довжини, який необхідно відсортувати. У результаті роботи цього алгоритму ми отримаємо відсортований масив введених елементів (табл. С1).

Таблиця С1

Вхідні та вихідні дані

Вхідні дані	Вихідні дані
Розмір вхідного масиву – $n$ . Масив з $n$ елементів, елементи – цілі числа ( масив $A[1.. n]$ – невідсортований ).	Впорядкований масив з $n$ елементів, елементи – цілі числа , кожен елемент масиву більший або дорівнює попередньому (масив $A[1.. n]$ ; $A[i] \leq A[i+1]$ ).

Алгоритм пірамідального сортування працює за два етапи:

I. Побудова сортуючого дерева;

Спочатку пірамідальне сортування повинне збудувати піраміду (англійський термін *heap*). ‘Піраміда’ – це дерево, в якому кожна вершина має значення менше або рівне, ніж її батьківська вершина. Крім того, використовують ‘бінарне дерево’, з додатковим обмеженням, що ніяка батьківська вершина не може мати більше двох нащадків; це дозволяє пірамідальному сортуванню виконуватися в межах початкового масиву, без додаткових затрат пам’яті. Кожну вершину проходимо в зворотньому порядку, якщо найбільша дочірня вершина більша від батьківської, переставляємо їх.

II. Просіювання елементів по сортуючому дереву.

Тепер пірамідальне сортування може виконувати сортування масиву. У цій ситуації функції пірамідального сортування дуже нагадують “зворотнє” сортування вибором, але “вибирання” є абсолютно необхідним; воно точно знає де є найбільші значення: на вершині піраміди. Перший елемент масиву переставляємо з останнім невідсортованим елементом, і піраміда перебудовується.

Як на I-ому, так і на II-ому етапах елементарна дія алгоритму полягає у вирішенні “сімейного” конфлікту: якщо найбільший із синів більший, ніж батько, то переставляються батько і цей син.

У результаті перестановки може виникнути новий конфлікт у тому трикутнику, куди переставлений батько. У такий спосіб можна говорити про конфлікт (роду) у піддереві з коренем у вершині  $i$ . Конфлікт роду вирішується послідовним вирішенням сімейних конфліктів проходом по дереву вниз. Конфлікт роду вирішено, якщо прохід закінчився ( $i > n \div 2$ ), або ж в результаті перестановки не виник новий сімейний конфлікт.

I етап – побудове сортуючого дерева - оформимо у виді рекурсивної процедури,

використовуючи визначення: якщо ліве і праве піддерева (  $T[2i]$  і  $T[2i+1]$  ) дерева  $T[i]$  є сортуючими, то для перебудови  $T[i]$  необхідно вирішити конфлікт роду в цьому дереві.

II етап - етап просіювання - для  $k$  від  $n$  до 2 повторюються наступні дії:

1. Переставити  $A[1]$  і  $A[k]$ ;
2. Побудувати сортуюче дерево початкового відрізка масиву  $A[1..k-1]$ , усунувши конфлікт роду в корені  $A[1]$ .

Далі наводимо детально описаний приклад пірамідального сортування масиву з восьми елементів.

Нехай ( 6, 5, 3, 1, 8, 7, 2, 4 ) – масив, який ми хочемо відсортувати від найменшого значення до найбільшого. Зауважимо, що для кроку “Побудова піраміди”: більші вершини не повинні розташовуватися нижче менших батьківських вершин. Їх переставляємо з батьківським, а тоді рекурсивно перевіряємо чи потрібне інше переставляння, щоб отримати більші.

Табл. С2 покроково пояснює етап побудови піраміди (сортуючого бінарного дерева).

Табл. С3 покроково пояснює етап сортування даних масиву на піраміді (сортуючому бінарному дереві).

## Побудова піраміди

Піраміда	Новий доданий елемент	Переставляння елементів
nul	6	
6	5	
6, 5	3	
6, 5, 3	1	
6, 5, 3, 1	8	
6, <b>5</b> , 3, 1, <b>8</b>		5, 8
<b>6</b> , <b>8</b> , 3, 1, 5		6, 8
8, 6, 3, 1, 5	7	
8, 6, <b>3</b> , 1, 5, <b>7</b>		3, 7
8, 6, 7, 1, 5, 3	2	
8, 6, 7, 1, 5, 3, 2	4	
8, 6, 7, <b>1</b> , 5, 3, 2, <b>4</b>		1, 4
8, 6, 7, 4, 5, 3, 2, 1		

## Сортування

Піраміда	Переставляння елементів	Вилучення елемента	Відсортований масив	Пояснення
<b>8, 6, 7, 4, 5, 3, 2, 1</b>	8, 1			Переставляння 8 і 1 для вилучення 8 з піраміди
<b>1, 6, 7, 4, 5, 3, 2, 8</b>		8		Вилучення 8 з піраміди і додавання до відсортованого масиву
<b>1, 6, 7, 4, 5, 3, 2</b>	1, 7		8	Переставляння 1 і 7, оскільки вони не є за порядком в піраміді
<b>7, 6, 1, 4, 5, 3, 2</b>	1, 3		8	Переставляння 1 і 3, оскільки вони не є за порядком в піраміді
<b>7, 6, 3, 4, 5, 1, 2</b>	7, 2		8	Переставляння 7 і 2 для вилучення 7 з піраміди
<b>2, 6, 3, 4, 5, 1, 7</b>		7	8	Вилучення 7 з піраміди і додавання до відсортованого масиву
<b>2, 6, 3, 4, 5, 1</b>	2, 6		7, 8	Переставляння 2 і 6, оскільки вони не є за порядком в піраміді
<b>6, 2, 3, 4, 5, 1</b>	2, 5		7, 8	Переставляння 2 і 5, оскільки вони не є за порядком в піраміді

<b>6, 5, 3, 4, 2, 1</b>	6, 1		7, 8	Переставляння 6 і 1 для вилучення 6 з піраміди
<b>1, 5, 3, 4, 2, 6</b>		6	7, 8	Вилучення 6 з піраміди і додавання до відсортованого масиву
<b>1, 5, 3, 4, 2</b>	1, 5		6, 7, 8	Переставляння 1 і 5, оскільки вони не є за порядком в піраміді
<b>5, 1, 3, 4, 2</b>	1, 4		6, 7, 8	Переставляння 1 і 4, оскільки вони не є за порядком в піраміді
<b>5, 4, 3, 1, 2</b>	5, 2		6, 7, 8	Переставляння 5 і 2 для вилучення 5 з піраміди
<b>2, 4, 3, 1, 5</b>		5	6, 7, 8	Вилучення 5 з піраміди і додавання до відсортованого масиву
<b>2, 4, 3, 1</b>	2, 4		5, 6, 7, 8	Переставляння 2 і 4, оскільки вони не є за порядком в піраміді
<b>4, 2, 3, 1</b>	4, 1		5, 6, 7, 8	Переставляння 4 і 1 для вилучення 4 з піраміди
<b>1, 2, 3, 4</b>		4	5, 6, 7, 8	Вилучення 4 з піраміди і додавання до відсортованого масиву
<b>1, 2, 3</b>	1, 3		4, 5, 6, 7, 8	Переставляння 1 і 3, оскільки вони не є за порядком в піраміді

<b>3, 2, 1</b>	3, 1		4, 5, 6, 7, 8	Переставляння 3 і 1 для вилучення 3 з піраміди
<b>1, 2, 3</b>		3	4, 5, 6, 7, 8	Вилучення 3 з піраміди і додавання до відсортованого масиву
<b>1, 2</b>	1, 2		3, 4, 5, 6, 7, 8	Переставляння 1 і 2, оскільки вони не є за порядком в піраміді
<b>2, 1</b>	2, 1		3, 4, 5, 6, 7, 8	Переставляння 2 і 1 для вилучення 2 з піраміди
<b>1, 2</b>		2	3, 4, 5, 6, 7, 8	Вилучення 2 з піраміди і додавання до відсортованого масиву
<b>1</b>		1	2, 3, 4, 5, 6, 7, 8	Вилучення 1 з піраміди і додавання до відсортованого масиву
			1, 2, 3, 4, 5, 6, 7, 8	Сортування завершене

Розглянутий процес пірамідального сортування можна представити у вигляді послідовності графів (рис. С1 та рис. С2).





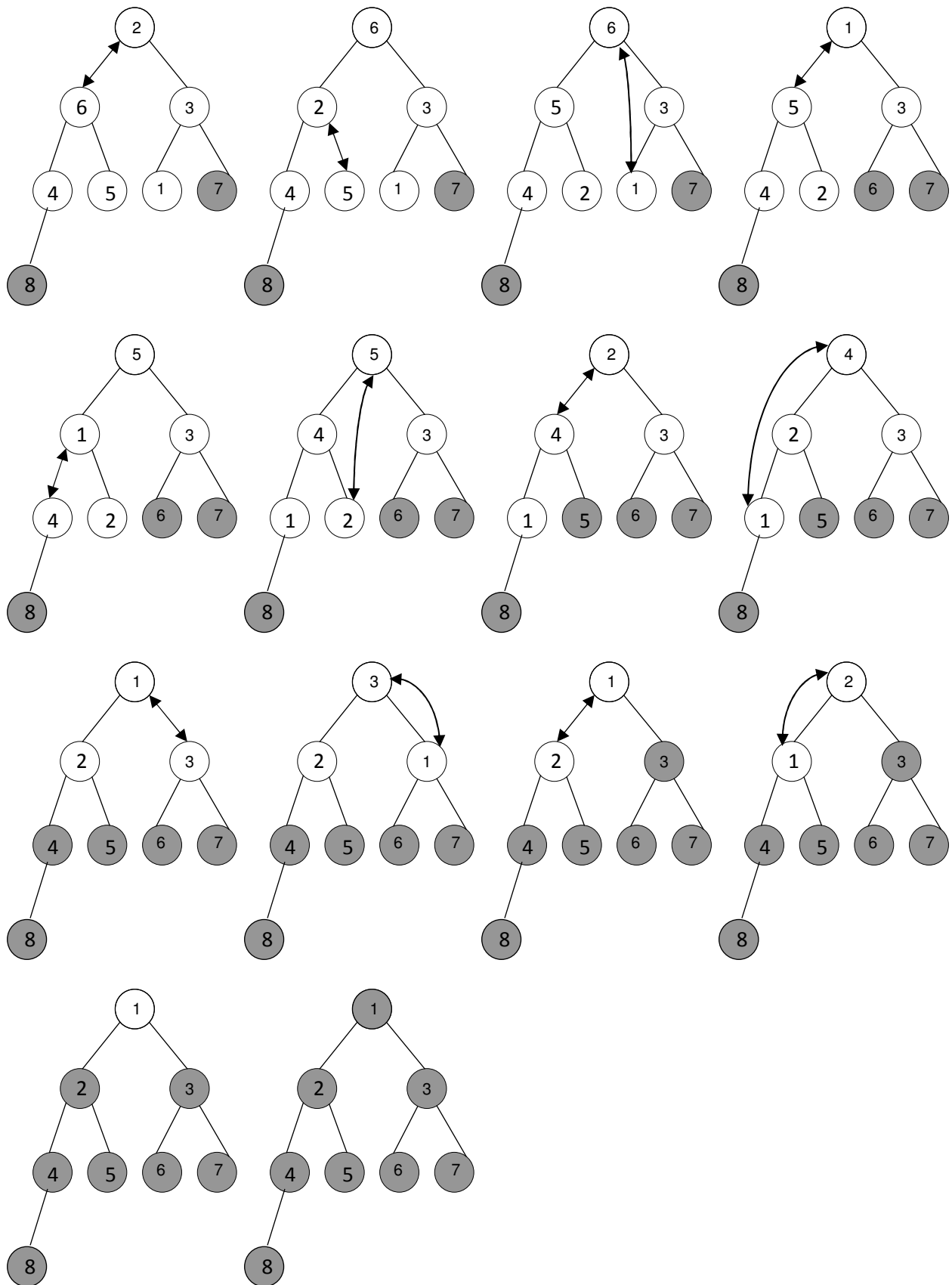


Рис.С2. Сортування