Пірамідальне сортування

Кормен, Лейзерсон, Рівест, Штайн. Алгоритми. 2-е вид.

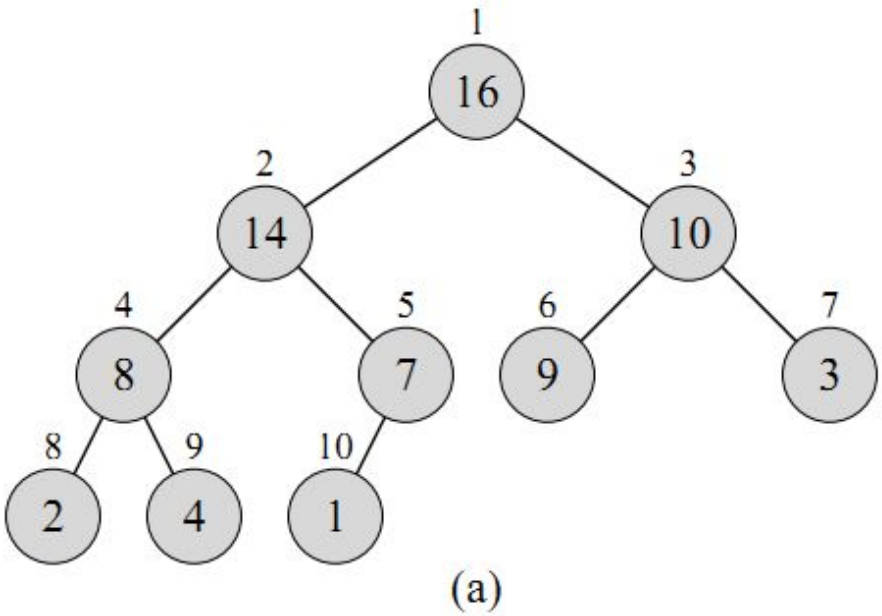
Пірамідальне сортування (Heap Sort)

- $\Theta(n \lg n)$ – як і час роботи алгоритму *merge sort*
- Не потребує додаткової пам'яті, як і *insertion sort*
- Найкращі особливості двох алгоритмів сортування, розглянутих раніше

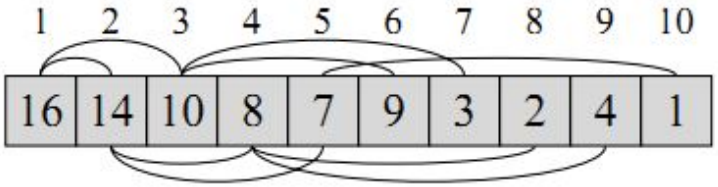
Піраміди

- ***Піраміда*** (binary heap) – структура даних, що являє собою об'єкт-масив, який можна розглядати як майже повне бінарне дерево.
- Кожен вузол цього дерева відповідає певному елементу масива.
- На всіх рівнях (можливо, крім останнього) дерево повністю заповнене
- Останній рівень заповнюється зліва направо до тих пір, поки в масиві не закінчаться елементи

Піраміди



(a)



(b)

Піраміди

- Масив A, що представляє піраміду є обє'ктом з двома атрибутами
  - length [A] – к-сть елементів масива
  - heap_size [A] – к-сть елементів піраміди, що містяться в масиві
- В корені дерева знаходиться A[1]
- Батьківський вузол для A[i] = A[i/2]
- Лівий дочірній вузол для A[i] = A[2i]
- Правий дочірній вузол для A[i] = A[2i+1]

Піраміди

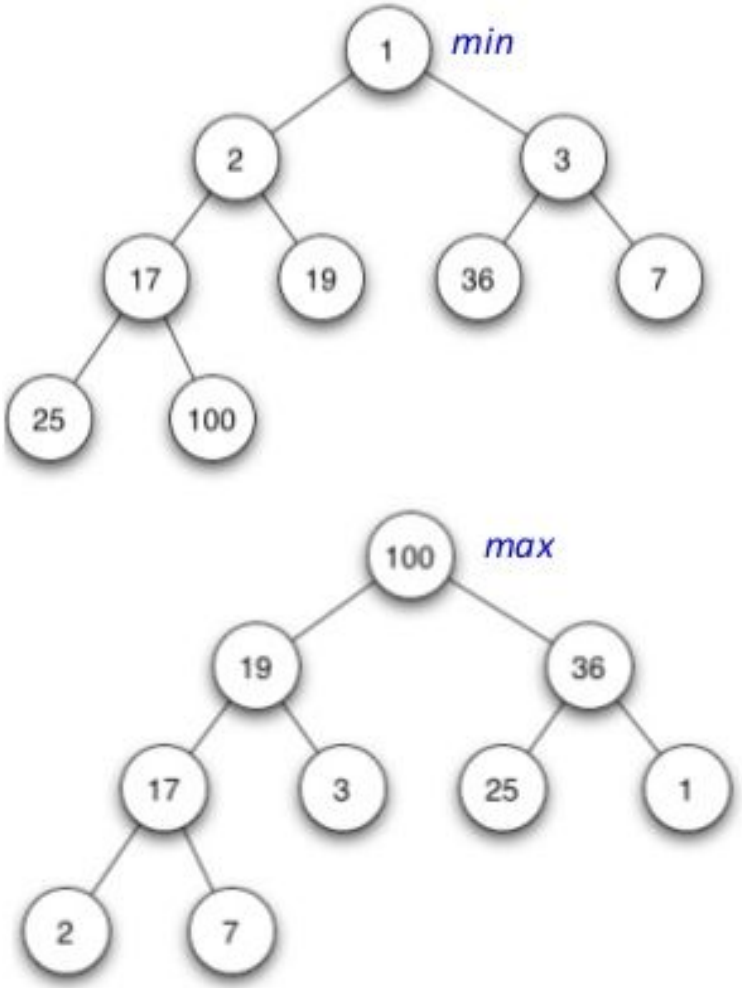
```
PARENT(i)
    return ⌊i/2⌋

LEFT(i)
    return 2i

RIGHT(i)
    return 2i + 1
```

Піраміди

- Неспадні (min-heap)
- Незростаючі (max-heap)



Піраміди

- Властивість незростаючих пірамід *(max-heap property)*

$$A[\text{PARENT}(i)] \geq A[i]$$

- Властивість неспадних пірамід *(min-heap property)*

$$A[\text{PARENT}(i)] \leq A[i]$$

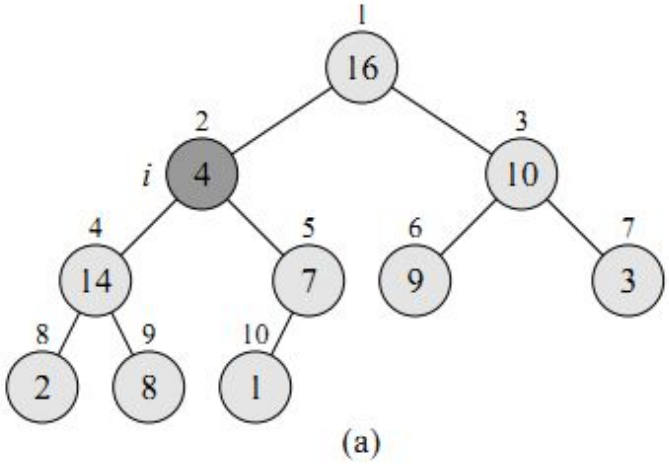
Піраміди

- Висота вузла піраміди – кількість ребер в найдовшому простому низхідному шляху від цього вузла до якогось із листів дереві
- Висота піраміди – висота її кореня
- Базові функції алгоритму сортування
  - Max_Heapify – *O(lg n)*
  - Build_Max_Heap – *O(n)*
  - Heapsort – *O(n lg n)*
  - Max_Heap_Insert, Heap_Extract_Max, Heap_Increase_Key, Heap_Maximum – *O(lg n)*

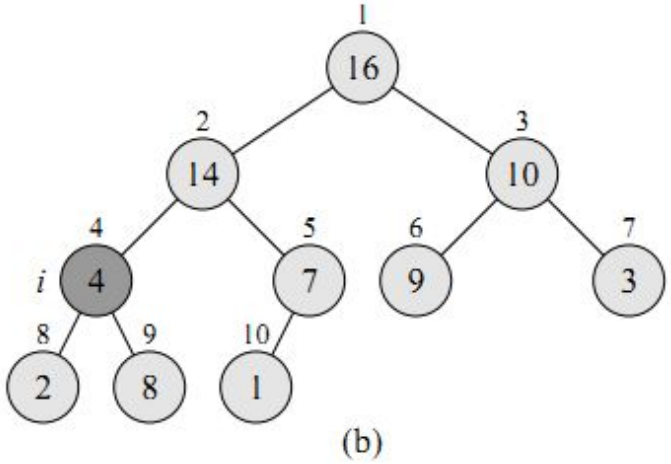
Піраміди

- 6.1-1. Чому дорівнює мінімальна і максимальна кількість елементів в піраміді висотою *h*?
- 6.1-2. Покажіть, що n-елементна піраміда має висоту [lg n].
- 6.1-4. Де в незростаючій піраміді може знаходитись найменший її елемент, якщо всі її елементи відрізняються за величиною?
- 6.1-5. Чи є масив з відсортованими елементами неспадною пірамідою?
- 6.1-6. Чи є послідовність (23,17,14,6,13,10,1,5,7,12) незростаючою пірамідою?

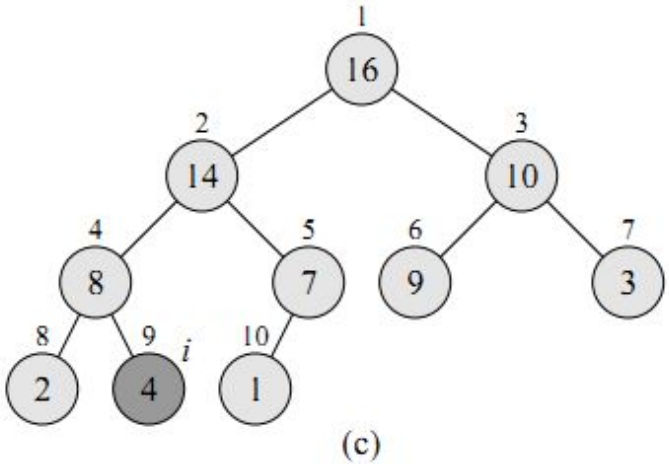
Підтримка властивості піраміди



(a)



(b)

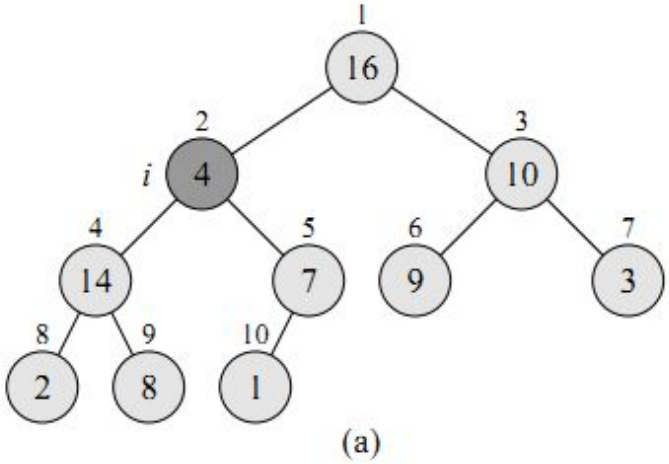


(c)

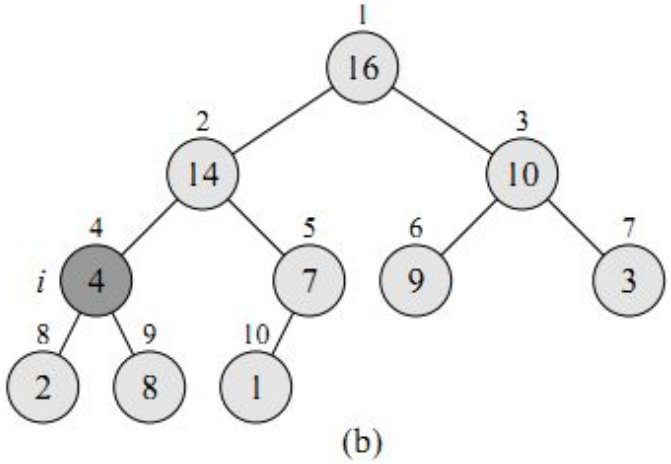
Підтримка властивості піраміди

```
MAX-HEAPIFY(A, i)
 1  l ← LEFT(i)
 2  r ← RIGHT(i)
 3  if l ≤ heap-size[A] and A[l] > A[i]
 4      then largest ← l
 5      else largest ← i
 6  if r ≤ heap-size[A] and A[r] > A[largest]
 7      then largest ← r
 8  if largest ≠ i
 9      then exchange A[i] ↔ A[largest]
10           MAX-HEAPIFY(A, largest)
```

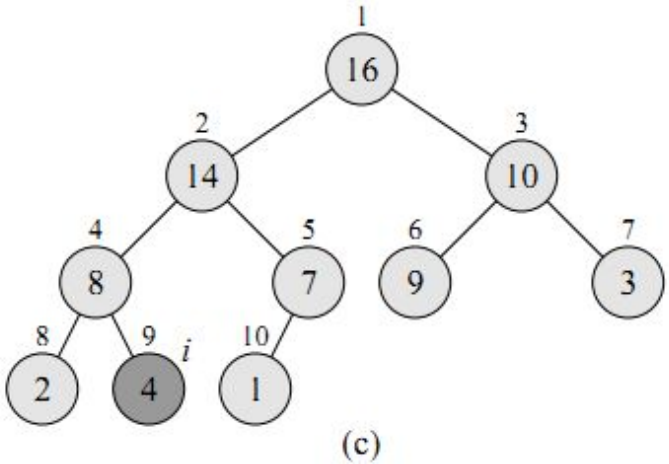
Підтримка властивості піраміди



(a)



(b)

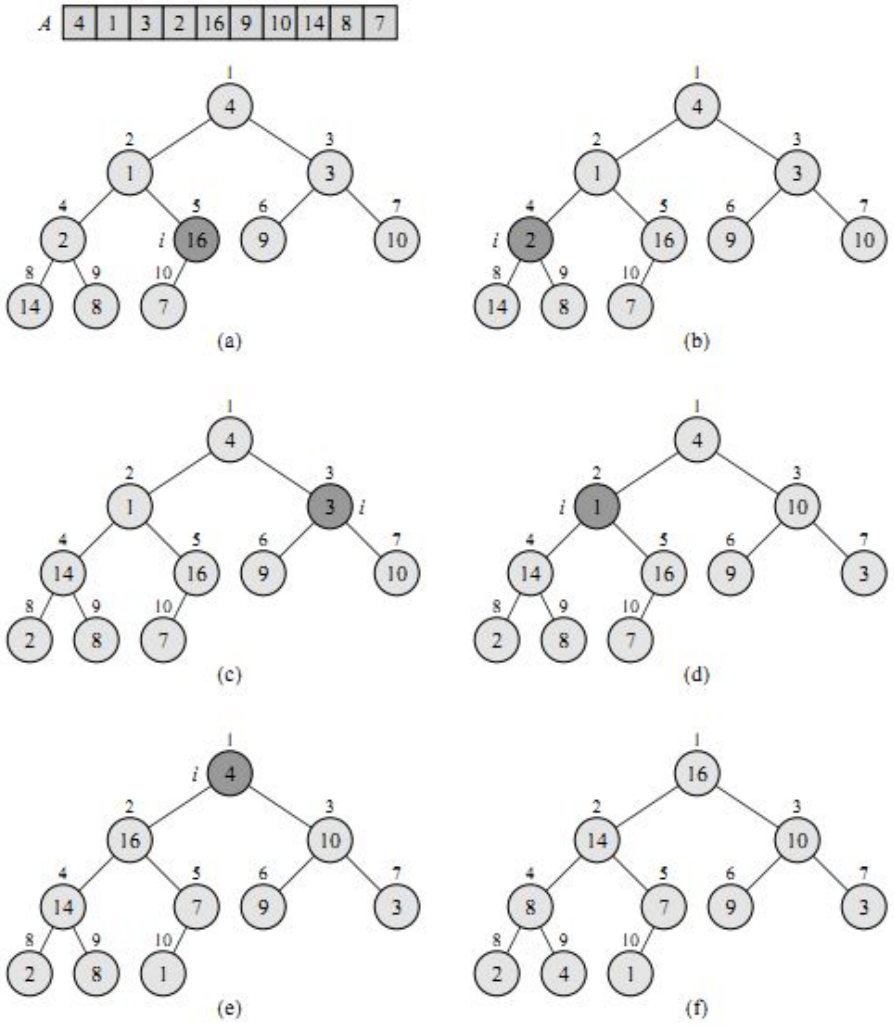


(c)

Підтримка властивості піраміди

- 6.2-1. Використовуючи в якості моделі рис. 6.2., проілюструйте роботу функції Max_Heapify (A,3) з масивом A=(27,17,3,16,13,10,1,5,7,12,4,8,9,0).
- 6.2-2. Використовуючи в якості відправної точки функцію Max_Heapify, напишіть псевдокод функції Min_Heapify(A,i), що виконує відповідні дії в неспадній піраміді. Порівняйте час роботи цих двох функцій.
- 6.2-3. Як працює функція Max_Heapify(A,i) у випадку, коли елемент A[i] більше своїх дочірніх елементів?
- 6.2-4. До чого призведе виклик функції Max_Heapify(A,i) при *i* > *heap_size [A]/2*?

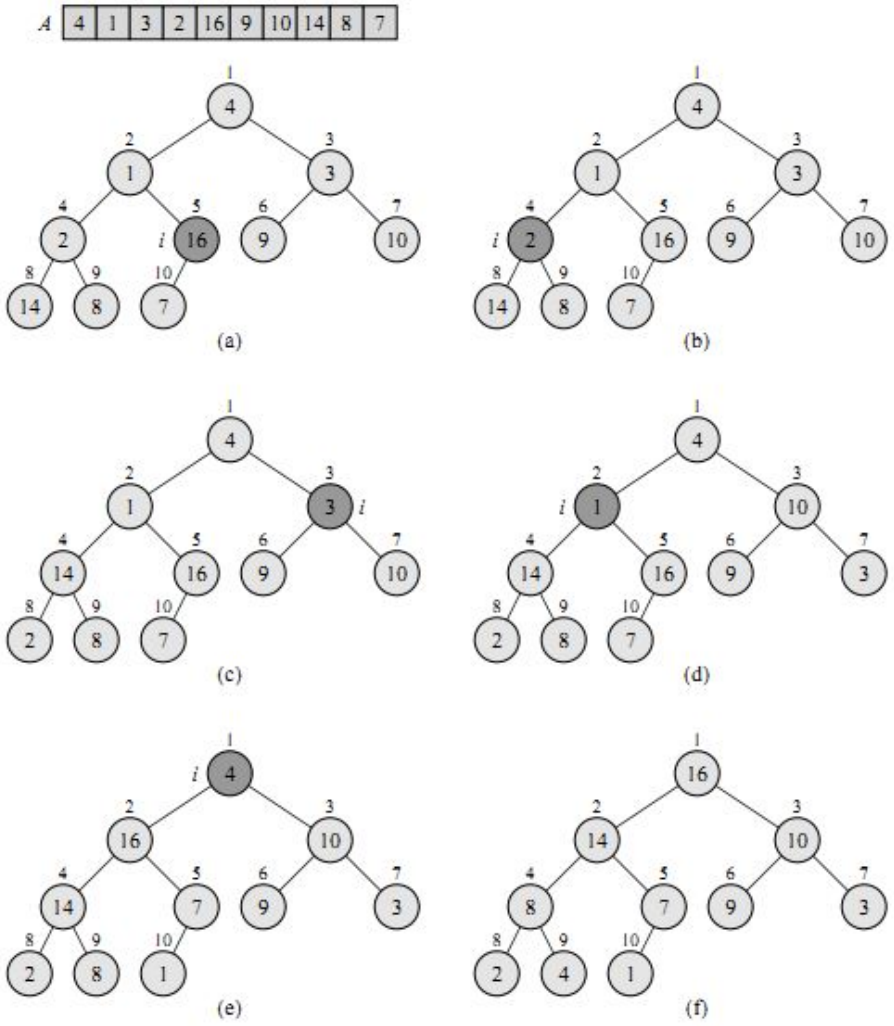
Створення піраміди



Створення піраміди

```
BUILD-MAX-HEAP(A)
1  heap-size[A] ← length[A]
2  for i ← ⌊length[A]/2⌋ downto 1
3        do MAX-HEAPIFY(A, i)
```

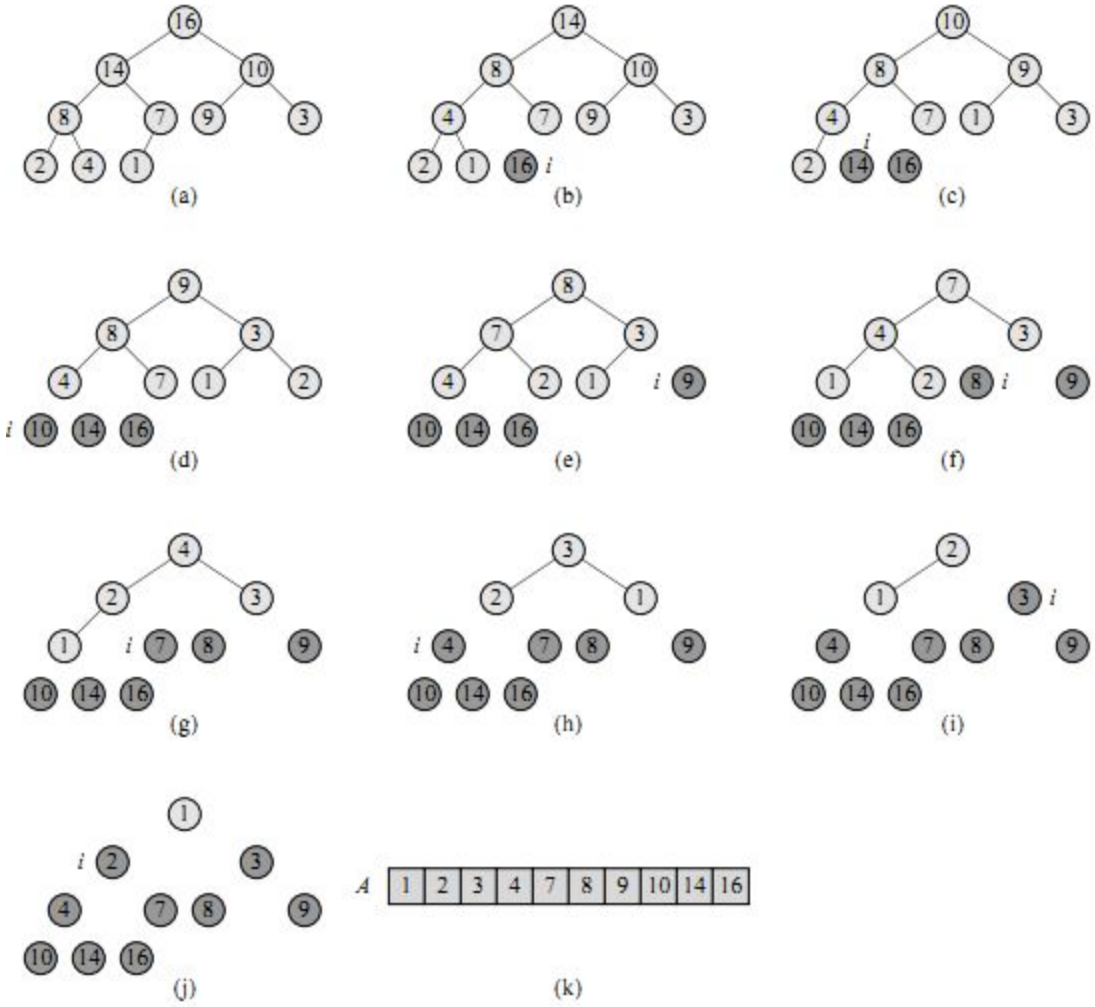
Створення піраміди



Створення піраміди

- 6.3-1. Використовуючи в якості моделі рис. 6.3, проілюструйте роботу функції Build_Max_Heap з вхідним масивом (5,3,17,10,84,19,6,22,9)
- 6.3-2. Чому індекс циклу *i* в рядку 2 функції Build_Max_Heap спадає від *length[A]/2* до 1, а не зростає від 1 до *length[A]/2*?

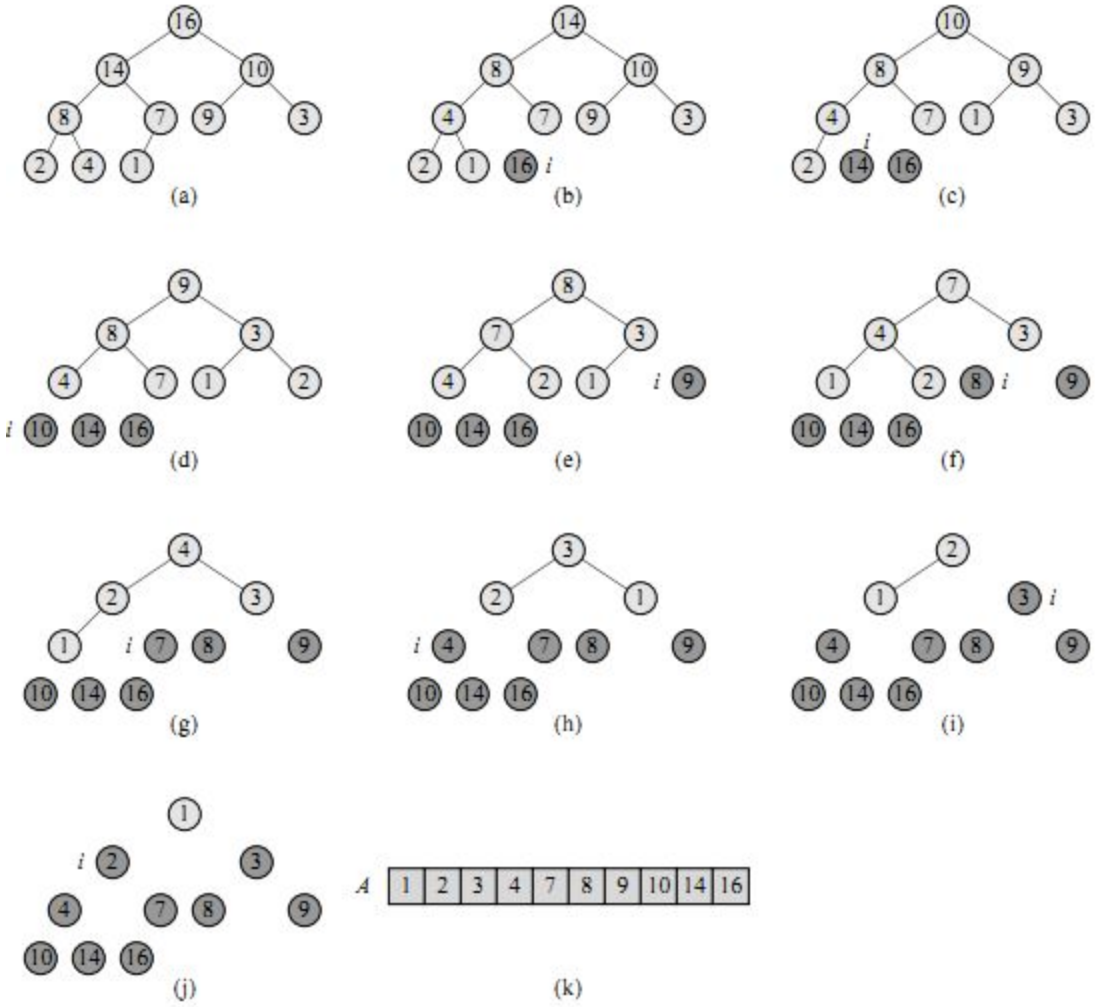
Алгоритм пірамідального сортування



Алгоритм пірамідального сортування

```
HEAPSORT(A)
1  BUILD-MAX-HEAP(A)
2  for i ← length[A] downto 2
3       do exchange A[1] ↔ A[i]
4          heap-size[A] ← heap-size[A] − 1
5          MAX-HEAPIFY(A, 1)
```

Алгоритм пірамідального сортування



Алгоритм пірамідального сортування

- 6.4-1. Використовуючи в якості моделі рис. 6.4, проілюструйте роботу функції Heapsort з вхідним масивом A=(5,13,2,25,7,17,20,8,4).

Алгоритм пірамідального сортування

Ваші запитання?