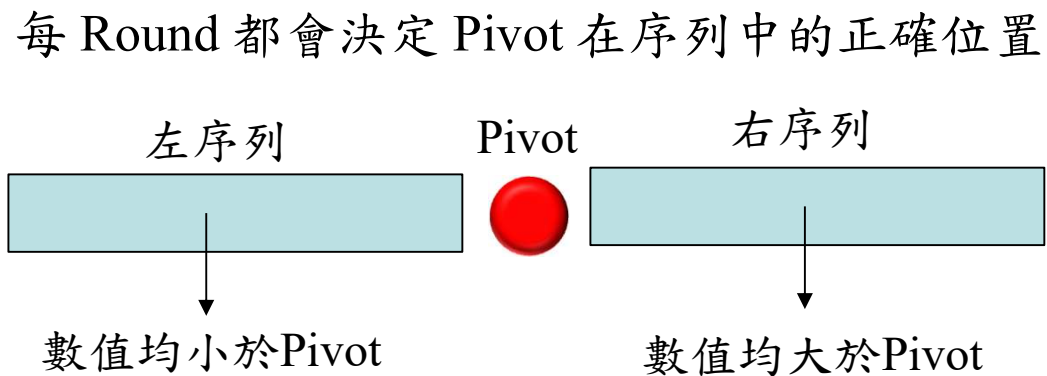


快速排序 (Quick Sort)

概念介紹

- Partition Exchange Sort
- 平均值行時間是所有排序中最佳的，故稱為快速排序
- 概念示意圖：



概念介紹

Step 1. 設定序列中第1個數字為 Pivot

Step 2. 維護 1 個指標 L ，初始指到序列第 2 個位置

維護 1 個指標 R ，初始指到序列最後 1 個位置

Step 3. 判斷 L 指到位址 與 R 指到位置之關係

Case 1: L 指到位址 = R 指到位置

➔ 直接比大小並結束 Round 的排序 (除了 Pivot 外，只剩 1 個元素)

Case 2: L 指到位址 \neq R 指到位置

(1) L 由左往右開始，找到第一個比 **Pivot** 大的值，STOP

(2) R 由右往左開始，找到第一個比 **Pivot** 小的值，STOP

(3) 判斷 L 指到位址 $<$ R 指到位置

Case 2-1: L 指到位址 $<$ R 指到位置

➔ R 位置的數值與 L 位置的數值對調

➔ 重複 (1)、(2) 和 (3)

Case 2-2: L 指到位址 \geq R 指到位置， R 位置的數值與 Pivot 對調

➔ Pivot 與 R 位置的數值對調

➔ 完成 Round 的排序 (分成左序列和右序列)

概念介紹

□ $A = [26, 5, 37, 1, 61, 11, 59, 15, 48, 19]$ ，請用快速排序法由小到大進行排序。

Round 1

Step 1. 設定序列中第1個數字為 Pivot

$[26, 5, 37, 1, 61, 11, 59, 15, 48, 19]$, Pivot = 26

Step 2. 維護 1 個指標 L ，初始指到序列第 2 個位置

維護 1 個指標 R ，初始指到序列最後1個位置

$[26, 5, 37, 1, 61, 11, 59, 15, 48, 19]$, Pivot = 26, $L = 2$, $R = 10$, Key(2) = 5, Key(10) = 19

Step 3. 判斷 L 指到位址 與 R 指到位置之關係

符合 Case 2: L 指到位址 $\neq R$ 指到位置 $\rightarrow L = 2, R = 10$

(1) L 由左往右開始，找到第一個比 Pivot 大的值，STOP

(2) R 由右往左開始，找到第一個比 Pivot 小的值，STOP

(3) 判斷 L 指到位址 $< R$ 指到位置

Case 2-1: L 指到位址 $< R$ 指到位置

$\rightarrow R$ 位置的數值與 L 位置的數值對調

\rightarrow 重複 (1), (2) 和 (3)

Case 2-2: L 指到位址 $\geq R$ 指到位置， R 位置的數值與 Pivot 對調

\rightarrow Pivot 與 R 位置的數值對調

\rightarrow 完成 Round 的排序 (分成左序列和右序列)

Example:

(1) $[26, 5, 37, 1, 61, 11, 59, 15, 48, 19]$, Pivot = 26, Key(L) = 37, Key(R) = 19

符合 Case 2-1. ($L = 3, R = 10$) $\rightarrow [26, 5, 19, 1, 61, 11, 59, 15, 48, 37]$ ，重複 Step 3

(2) $[26, 5, 19, 1, 61, 11, 59, 15, 48, 37]$, Pivot = 26, Key(L) = 61, Key(R) = 15

符合 Case 2-1. ($L = 5, R = 8$) $\rightarrow [26, 5, 19, 1, 15, 11, 59, 61, 48, 37]$ ，重複 Step 3

(3) $[26, 5, 19, 1, 15, 11, 59, 61, 48, 37]$, Pivot = 26, Key(L) = 59, Key(R) = 11

符合 Case 2-2. ($L = 7, R = 6$) $\rightarrow [11, 5, 19, 1, 15, 26, 59, 61, 48, 37]$ ，完成 Round 排序

\rightarrow Round 1 : $[11, 5, 19, 1, 15, 26, 59, 61, 48, 37]$ ，左序列 = $[11, 5, 19, 1, 15]$ ，右序列 = $[59, 61, 48, 37]$

概念介紹

□ $A = [26, 5, 37, 1, 61, 11, 59, 15, 48, 19]$ ，請用快速排序法由小到大進行排序。

Round N

Step 1. 設定序列中第1個數字為 Pivot

$[1, 5]$, Pivot = 1

Step 2. 維護 1 個指標 L ，初始指到序列第 2 個位置

維護 1 個指標 R ，初始指到序列最後1個位置

$[1, 5]$, Pivot = 26, $L = 2$, $R = 2$, Key(2) = 5, Key(2) = 5

Step 3. 判斷 L 指到位址 與 R 指到位置之關係

符合 Case 1: L 指到位址 = R 指到位置 $\rightarrow L = 2, R = 2$

直接比大小並結束 Round 的排序

\rightarrow Round N : $[1, 5]$

概念介紹

□ $A = [26, 5, 37, 1, 61, 11, 59, 15, 48, 19]$ ，請用快速排序法由小到大進行排序。

Round $N+1$

Step 1. 設定序列中第1個數字為 Pivot

$[19, 15], \text{Pivot} = 1$

Step 2. 維護 1 個指標 L ，初始指到序列第 2 個位置

維護 1 個指標 R ，初始指到序列最後1個位置

$[19, 15], \text{Pivot} = 19, L = 2, R = 2, \text{Key}(2) = 15, \text{Key}(2) = 15$

Step 3. 判斷 L 指到位址 與 R 指到位置之關係

符合 Case 1: L 指到位址 = R 指到位置 $\rightarrow L = 2, R = 2$

直接比大小並結束 Round 的排序

\rightarrow Round N : $[15, 19]$

Heapsort Algorithm

Heapsort Algorithm

HEAPSORT(*A*)

```

1 BUILD-MAX-HEAP(A)
2 for i = A.length downto 2
3   exchange A[1] with A[i]
4   A.heap-size = A.heap-size - 1
5   MAX-HEAPIFY(A, 1)
```

BUILD-MAX-HEAP(*A*)

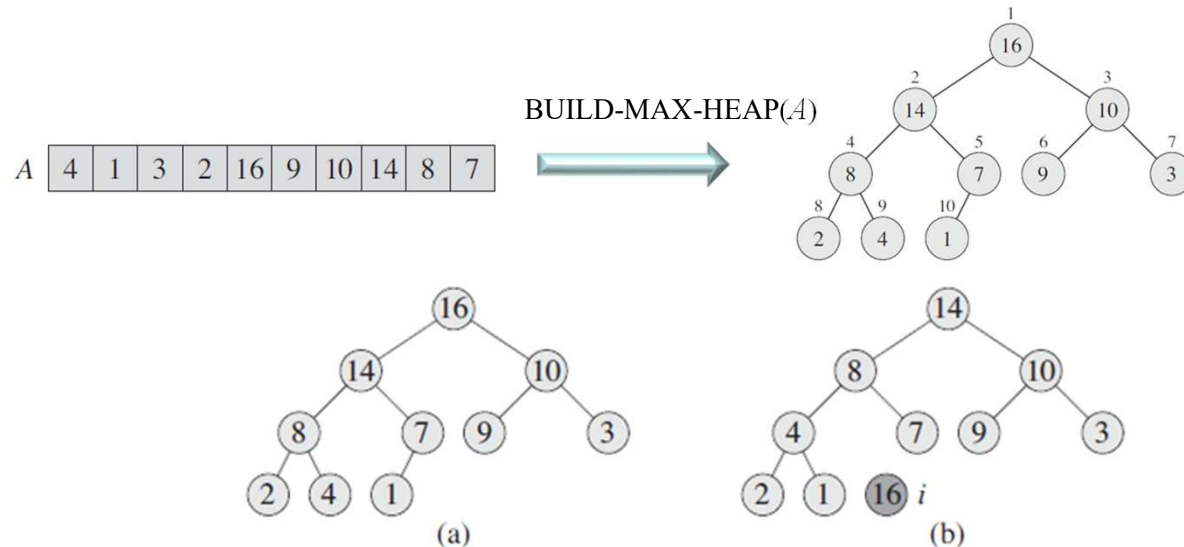
```

1 A.heap-size = A.length
2 for i =  $\lfloor A.length/2 \rfloor$  downto 1
3   MAX-HEAPIFY(A, i)
```

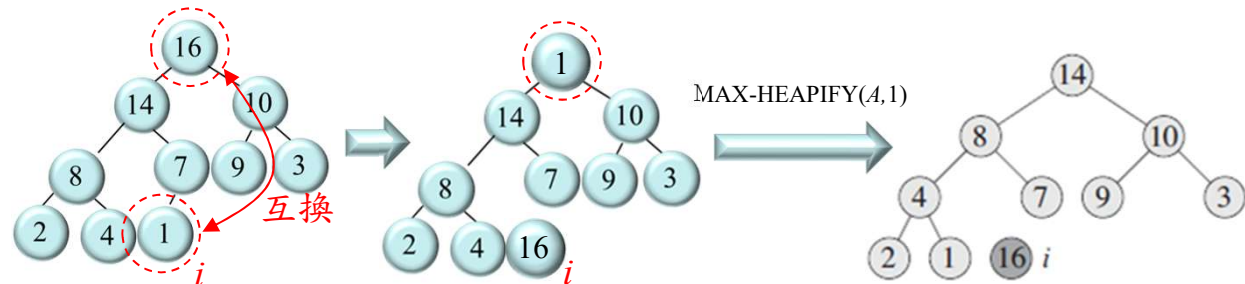
MAX-HEAPIFY(*A*, *i*)

```

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



Round 1



Heapsort Algorithm

HEAPSORT(A)

```

1 BUILD-MAX-HEAP( $A$ )
2 for  $i = A.length$  downto 2
3   exchange  $A[1]$  with  $A[i]$ 
4    $A.heap-size = A.heap-size - 1$ 
5   MAX-HEAPIFY( $A, 1$ )
    
```

BUILD-MAX-HEAP(A)

```

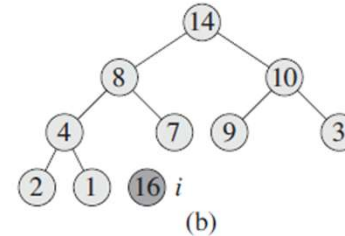
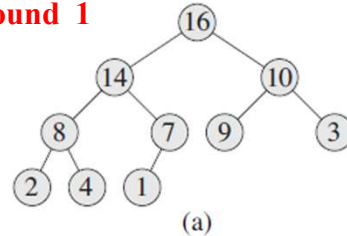
1  $A.heap-size = A.length$ 
2 for  $i = \lfloor A.length/2 \rfloor$  downto 1
3   MAX-HEAPIFY( $A, i$ )
    
```

MAX-HEAPIFY(A, i)

```

1  $l = LEFT(i)$ 
2  $r = RIGHT(i)$ 
3 if  $l \leq A.heap-size$  and  $A[l] > A[i]$ 
4    $largest = l$ 
5 else  $largest = i$ 
6 if  $r \leq A.heap-size$  and  $A[r] > A[largest]$ 
7    $largest = r$ 
8 if  $largest \neq i$ 
9   exchange  $A[i]$  with  $A[largest]$ 
10  MAX-HEAPIFY( $A, largest$ )
    
```

Round 1



Round 2

