

Quick Sort

Two methods to avoid the worst case

109201521 彭康彧 108303202 趙元彰 109502575 許明瑞

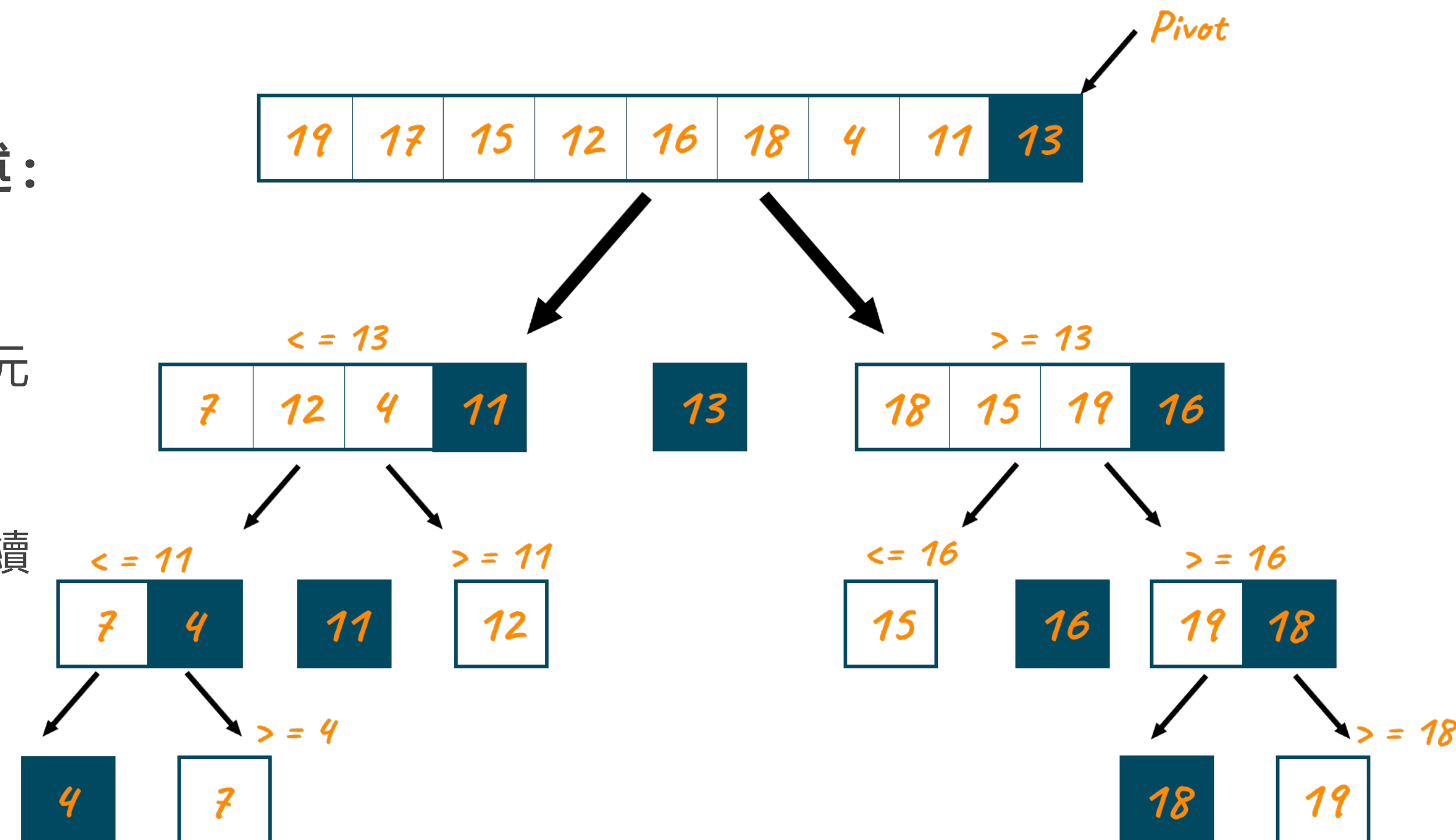
Quick Sort

快速排序演算法使用分治解題策略，其做法如下所述：

分割：選一個元素p當作中樞(pivot)元素將陣列分割為2部份：SP及LP，其中SP (smaller part)包含所有小於p的元素；而LP(larger part)則包含所有大於或等於p的元素。

克服：完成陣列分割(partition)之後，快速排序演算法持續遞迴地(recursively)進行SP部份與LP部份的元素排序。

合併：最後再將SP、p及LP合併即可完成排序。





Time Complexity of QuickSort

Best Case - $O(n \log n)$

Worst Case - $O(n^2)$

Average Case - $O(n \log n)$



Time Complexity of QuickSort

Best Case - $O(n \log n)$

~~Worst Case - $O(n^2)$~~

Average Case - $O(n \log n)$

1.

步驟如下：

(1) 首先我們發現對 Quick Sort 來說最壞的狀況其實就是已經排序好的狀況，或順序相反。因此只要檢查是否是排序好的狀況，並在進入遞迴之前 return。

(2) 我們也發現在若是已經排好的狀態，會發生前指標或後指標不動（稱 A 指標），但另外一個指標（稱 B 指標）會一直移動到 A 指標的另一側。

(3) 若是排序好的list，我們將直接return; 排序相反的就先將內容相反在return，否則就進入遞迴。

9	8	7	6	5	4	3	2	1
---	---	---	---	---	---	---	---	---

▲ 排序相反的狀況

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

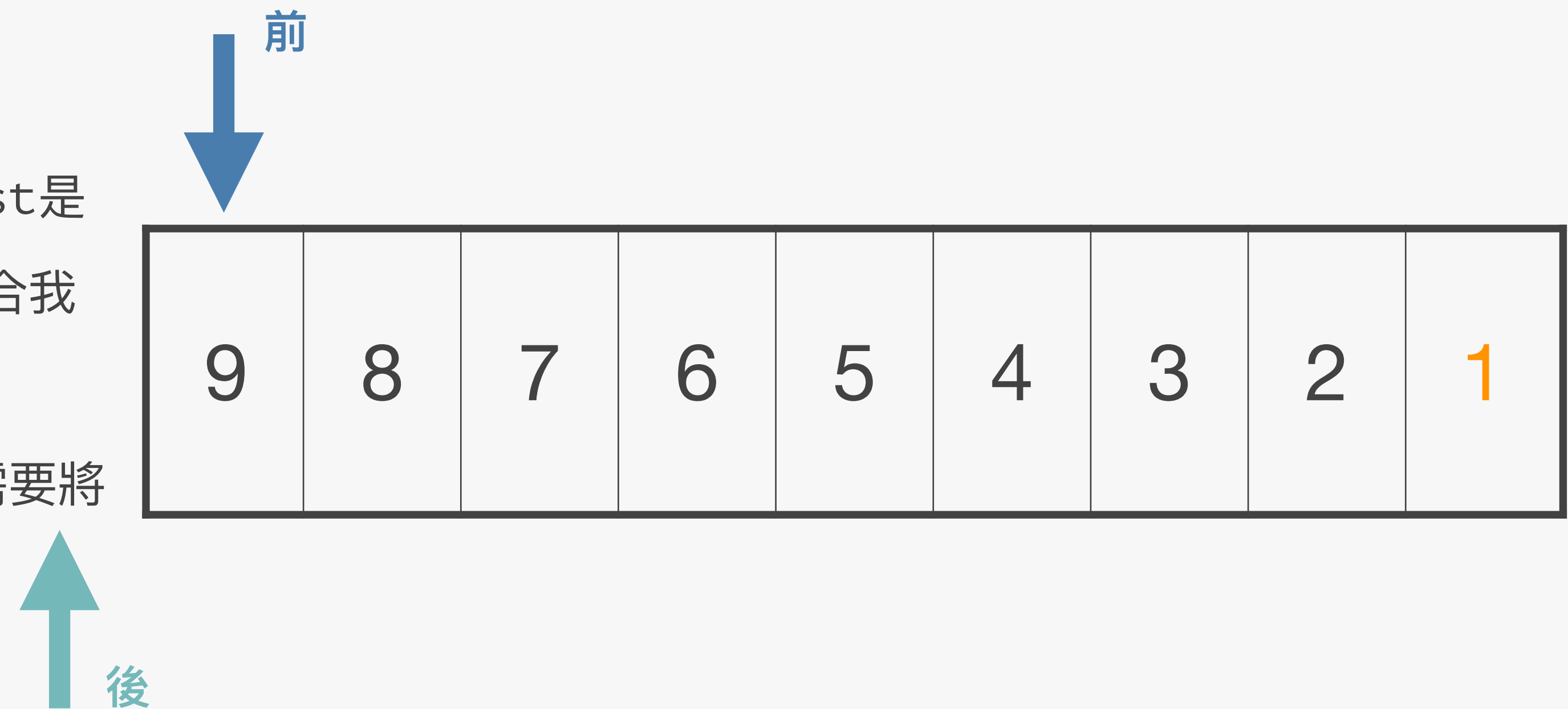
▲ 已排序好的狀況

1.

範例說明：

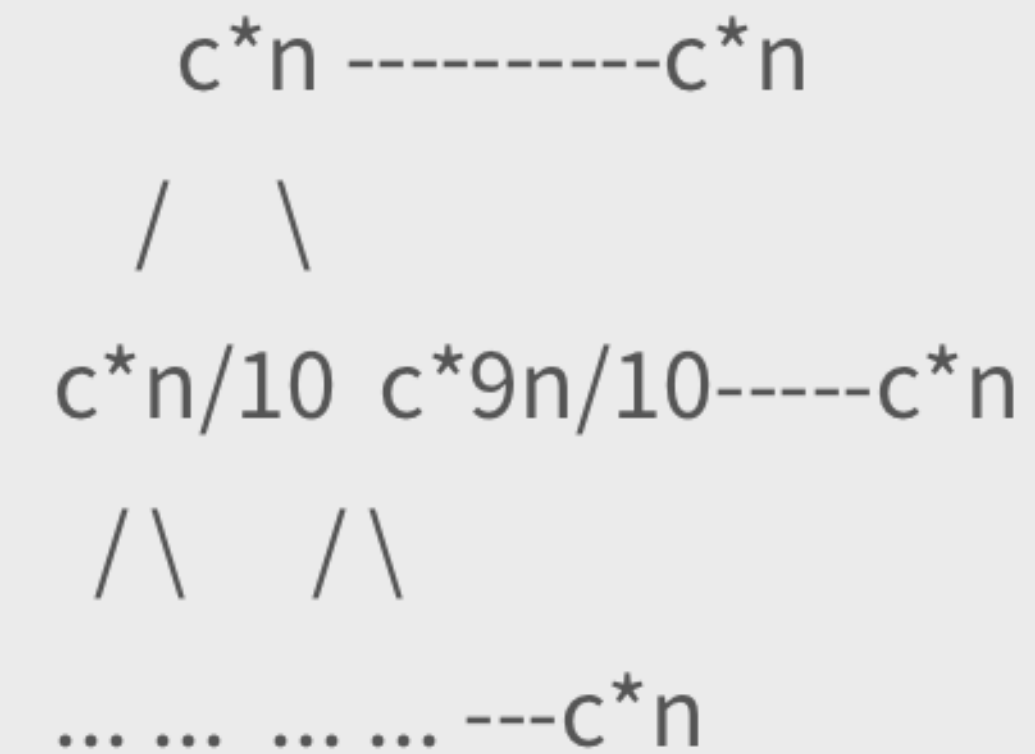
例如：若A指標為前指標，B指標為後指標，而list是排序相反，後指標會移動到前指標前面。此時就符合我們先前講的條件。

也就是說可以知道這是排序相反的序列，我們只需要將內容相反即可return，總計時間僅有 $O(n)$ 。



2.

- (1) 令 $\text{middle} = (\text{left} + \text{right}) / 2$
- (2) 比較 $A[\text{left}]$ 、 $A[\text{middle}]$ 與 $A[\text{right}]$ 這三筆資料，排出中間值。
- (3) 將此中間值再與 $A[\text{left}]$ 做交換
- (4) 讓現在新的 $A[\text{left}]$ 作為 pivot 如果 pivot 的位置恰好在 n 筆資料中的 $n/10$ 與 $9n/10$ 之間：
最後還是可以收斂到 $O(n \log n)$



$$T(n) = c^*n + T(n/10) + T(9n/10)$$