

前言：

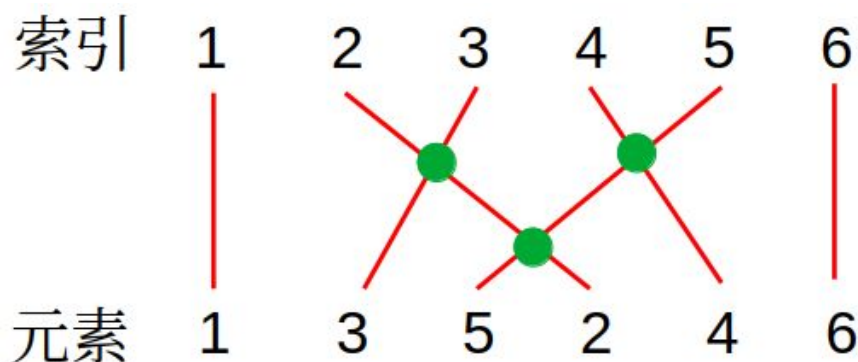
第二周的課程繼續探討分治法的概念：

1. 將問題拆成較小的子問題。
2. 利用遞迴持續拆解問題。
3. 將子問題的解組合回原問題的尺寸。

作業為利用分治法計算給定資料的 inversion 數量。

Inversion 的計算與分治法：

Inversion 的定義為：陣列中若有兩元素的索引值 $i < j$ ，但 $A[i] > A[j]$ ，則稱這樣的索引對為 inversion，例如某陣列 $[1, 3, 5, 2, 4, 6]$ ，inversion 的數量為 3，索引對分別為 $\{(3, 2), (5, 2), (5, 3)\}$ 。



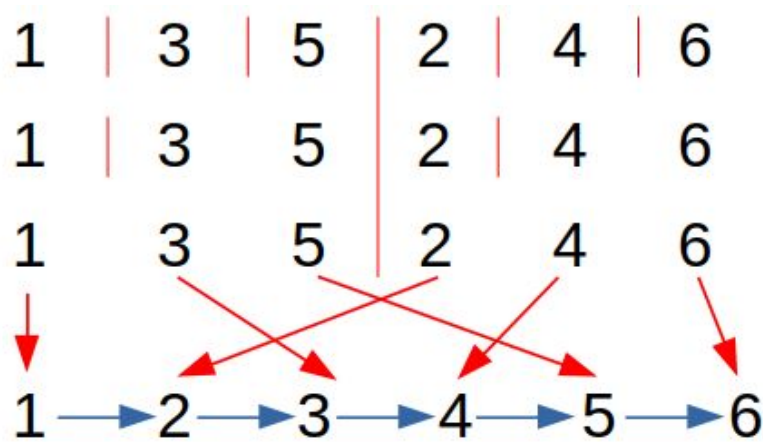
那計算 inversion 有什麼用處呢？可以想像這個陣列是某一個用戶的 page rank list，經由比對 inversion 的數量，計算與該用戶最相近的使用者，系統就可以推薦其他相近用戶最近瀏覽的網頁，或是購買的商品等等。

若以 brute force 求解，需兩個迴圈比對每個元素，因此複雜度為 $O(n^2)$ 。那該如何加速計算的過程？其實可以利用合併排序的概念，將 inversion 問題拆成兩個子問題，每個子問題先個別計算，再合併兩個子問題計算左右陣列的 split inversion，最後將三個結果數字相加，即可得到答案。

演算法流程：

以同陣列 [1, 3, 5, 2, 4, 6] 為例：

1. 首先將問題拆成 2 個子問題 [1, 3, 5] 及 [2, 4, 6]。
2. 利用遞迴，持續將問題拆成 6 個子問題。
3. 因為每個子問題都只有一個元素，沒有 inversion，進行合併。
4. 合併後仍然沒有 inversion，再次進行合併。
5. 合併成 [1, 3, 5]、[2, 4, 6] 之後，要再合併時發現 inversion，進行排序的同時計算 inversion：
 - a. 左側元素 1 為最小，將 1 取出。([3, 5]、[2, 4, 6])
 - b. 右側元素 2 為最小，將 2 取出，因左側還有兩個元素，代表 2 跟 [3, 5] 皆為 inversion，count 加 2。([3, 5]、[4, 6])
 - c. 左側元素 3 為最小，將 3 取出 ([5]、[4, 6])
 - d. 右側元素 4 為最小，將 4 取出，count 加 1。([5], [6])
 - e. 最後將 5 跟 6 依序取出完成計算，總 count 為 3。



演算法正確性：

假設左半陣列 A 及右半陣列 B 各自皆已排序，x 為左半陣列 A 的元素；y 為右半陣列 B 的元素，兩陣列合併後輸出陣列為 D，可分為兩個情況考慮：

1. x 比 y 還要早被複製到 D 陣列中 → 代表 x 小於 y，沒有 inversion。
2. x 比 y 晚被複製到 D 陣列中 → 代表 x 大於 y，x, y 為 inversion pair。y 被複製時，左半部剩餘的元素都比 y 大，因此其長度為 y 的 split inversion 數量。

時間複雜度：

該演算法基本上與合併排序無異，只是在其基礎上增加 inversion 的計算而已，所以時間複雜度為 $O(n\log n)$

作業——以分治法方式計算給定資料 inversion 的數量：**問題描述：**

資料中共有 1~100,000 的整數，以某種順序排列，無重複的數字。請計算該資料 inversion 的數量。

解題方法：

利用合併排序的概念即可求出答案，一共 2,407,905,288 個。