tags: Algorithms

Algorithms 2020 Spring HW2

Handwritten

Author: 90899201Y tony20715 黃悟淳(選讀)

1. Quick Sort Attack

(1)

- The time complexity for the worst, the best, and the average case is $\Theta(n^2)$, $\Theta(n*log(n))$, and $\Theta(n*log(n))$, respectively.
- Proof: 設原題目數列 A 長度為 n ，為 $A[0..(n-1)]$
  i. Worst Case 每次選的pivot number皆將原先有 x 個數的數列切分成 $x-1$ 個數的數列和一個空的數列，亦即每次都選到數列中的最大值或最小值，即為最差狀況。 在此狀況下，總共需要做 $n-1$ 次迴圈，其中在第 i 次迴圈時 $(1 \leq i \leq (n-1))$ ，需做 $i-1$ 個與pivot number的比較。故總比較次數 $T(n)=\sum_{i=1}^{n-1}\sum_{j=1}^{i-1}\Theta(1)=\sum_{i=1}^{n-1}(i-1)\Theta(1)=\frac{(n-2)(n-1)}{2}\Theta(1)=\Theta(n^2)$。
  ii. Best Case 對於每一個長度為 x 的待排序數列，每次都選到第 $\frac{x}{2}$ 大的數作為pivot number，使得原數列被對分為兩個長度為 $\lfloor\frac{x-1}{2}\rfloor$、$\lceil\frac{x-1}{2}\rceil$ 的數列，亦即利用divide and conquer的概念將原問題對分成2個大小為 $x/2$ 的小問題。利用Master Theorem: $T(n)=2T(n/2)+f(n)$ ，其中 $f(n)=PARTITION(a,p,r)$。\Theta(n*log_ba)=\Theta(n*log_22)=\Theta(n)f(n)=PARTITION(a, p, r)=\Theta(p-r)(程式碼僅一層for\ j =p \to (r-1)的迴圈)。又n=p-r+1，故f(n)=\Theta(n-1)=\Theta(n)。\Theta(n*log_ba)= f(n) = \Theta(n)，即$MasterTheorem$的$case2$，即T(n)=\Theta(n*log(n))$
  iii. Average Case 假設每次長度為 x 的數列都會被pivot number拆分成長度為 $\frac{x}{10}$ 和 $\frac{9x}{10}$ 的兩數列，即 $T(n)=T(\frac{n}{10})+T(\frac{9n}{10})+f(n)$，其中 $f(n)=\Theta(n)$ 同Best Case。每次都屬於 $\frac{x}{10}$ 的數列會在$log_{10}n$次遞迴後抵達$\Theta(1)$，而每次都屬於$\frac{x}{10}$ 的數列會在$log_{10/9}n$次後遞迴抵達$\Theta(1)$。每次遞迴都要執行一次 $f(n)=\Theta(n)$ ，故 $T(n)<log_{10/9}n*\Theta(n)=O(n*log(n))$，且 $T(n)>log_{10}n*\Theta(n)=\Omega(n*log(n))$。由夾擠定理，$T(n)=\Theta(n*log(n))$。

(2)

```
int *attack(int n){
    int *arr = malloc(n * sizeof(int));
    for (int i=0; i<n; i++)
        *(arr+i) = i+1;
    return arr;
}
```

產生一個首項為1，公差為1，項數為n的數列填入attack陣列。由於題目的程式碼以每次數列的最後一個值當作pivot number，並要進行由小到大排列。若每次程式拿到的pivot number都是最大值，則每遞迴都會將原長x數列分成長(x-1)的數列和空數列，以致必須進行(n-1)次遞迴，符合(1)小題的worst case scenario。

2. Bubble Sort

(1)

- (a) k。 根據程式碼，bubble sort將把數列由小至大排列。因此，如果 a_i 在數列中是第k小的數，排序完以後它的index應該是k。
- (b) $x+y$。 若數列為由左至右排列，比 a_i 小的要到左邊，比 a_i 大的要到右邊，所以只有比 a_i 小又在右邊的數 (y個) 和比 a_i 大又在左邊的數 (x個) ，要和 a_i 進行交換(swap)。每一次I迴圈 (`for i = 1 to A.length - 1` 迴圈) a_i 最多只能

swap一次，所以 a_i 必須做 $x+y$ 次交換，才能把 x 個比它大又在左邊的數換到右邊，並把 y 個比它小又在右邊的數換到左邊

- c $x+y$。 inversion數就是順序最終要被交換的pair。已知有 x 個比 a_i 大又在左邊的數，y 個比 a_i 小又在右邊的數，這些數和 a_i 是要被交換的pair，所以總共的inversion數即為 $x+y$。

(2)

- (a) A swapping occurs between the current number and its previous number if and only if the current number is smaller than the previous number. Namely, there is an inversion which will be eliminated after the swapping being executed. Those pairs which component numbers are not adjacent to each other will not be eliminated since the component numbers will not be compared during the process. Therefore, an executed swapping is equivalent to a modification of a pair of numbers letting them be arranged in the right order, or namely, an elimination of an inversion.
- (b) 首先，正確排序的數列，亦即完成bubble sorting的數列，其inversion數為0。其次，每進行一次swapping，inversion數必定會減少1。如果bubble sort總共進行了n次swapping，才成為正確排序的數列，換言之原inversion減去 $n*1$ 後會等於0，以符號表示，即為 $I(A)-n*1=0$，得 $I(A)=n$。