COT5405 Homework 1 Solutions

Maksim Levental

August 31, 2014

2.4 (a) (2,1), (3,1), (8,1), (6,1), (8,6)

(b) Reverse sorted, i.e. $\{n, n-1, \cdots, 1\}$. The number of inversions is $n-1$ for 1 because there are $n-1$ elements in the array which are larger than 1 but preceed it in the array, $n-2$ for 2 because there $n-2$ elements which are larger than 2 (exceptions are 2 and 1) but preceed it, and so on. So for $i = 1, 2, \cdots, n$ the number of inversions induced is $n-i$. In sum $\sum_{i=1}^{n} n - i = n\sum_{i=1}^{n} 1 - \sum_{i=1}^{n} i = n^2 - \frac{n(n+1)}{2} = n^2 - \frac{(n^2+n)}{2} = \frac{n(n-1)}{2}$

(c) Call an inversion of type (j, i) induced by an element $A[j]$ if j is such that $A[j] < A[i]$ and $i < j$ and let $|(j, i)|$ be the number of such inversions. Then $|(j, i)|$ is the number of swaps that will have to be performed on element $A[j]$ before it is in its proper position. To see that this is the case note that all (j, i) inversions persist through the sorting process, up until $A[j]$ is sorted, since for all $i < j$ element $A[i]$ will be inserted into the sorted portion of the array prior to $A[j]$. Therefore upon inserting $A[j]$ there will still be $|(j, i)|$ inversions and therefore $|(j, i)|$ swaps. Consequently $\sum_j |(j, i)|$ the total number of inversions in the array is the total number of swaps performed by insertion sort, i.e. directly proportional insertion sort's running time.

(d) Modify merge sort such that when function returns from the two recursive calls, when the "merging" is done, it counts the number of elements in the "left" array each time an element is chosen from the front of the "right" array (after comparison between the leading elements of both arrays). The quantity of elements in the "left" array each time an element from the front of the right array is chosen is by definition the number of elements in the original array that were greater than that chosen element and yet preceeded it. Furthermore there is no double counting because once a merge happens 2 elements in the merge array are never compared again. Take for example the array $[8, 7, 6, 5, 4, 3, 2, 1]$ and suppose the recursion bottoms out at 4 elements. Then the first recursion returns $[5, 6, 7, 8]$ and $[1, 2, 3, 4]$. The merging then selects each of the four elements from the "right" array $[1, 2, 3, 4]$ since each of the elements in the "left" array $[5, 6, 7, 8]$ is greater than each in the "right". Manifestly there are 4 inversion per element in the "right" array - 1 for each element in the "left" array, and so the total is 16.

3.2

AB	Oo	$\Omega\omega$	Θ
$\lg^k n n^\epsilon$	$yesyes$	$yesyes$	yes
$n^k c^n$	$yesyes$	$yesyes$	yes
$\sqrt{n} n^{\sin n}$	$yesyes$	$yesyes$	yes
$2^n 2^{n/2}$	$yesyes$	$yesyes$	yes
$n^{\lg c} c^{\lg n}$	$yesyes$	$yesyes$	yes
$\lg(n!) \lg(n^n)$	$yesyes$	$yesyes$	yes