CSCI-570: Homework # 2

Due on Friday, September 12 , 2014

Saket Choudhary
skchoudh@usc.edu
2170058637

Contents

HW2

(2: Ch#2 Ex#3)

$f1 = n^2.5, f2 = \sqrt{2n}, f3 = n + 10, f4 = 10^n, f5 = 100^n, f6 = n^2 logn$
Consider the square of f_i:

$f1' = n^5; f2 = 2n; f3 = (n + 10)^2; f4 = 10^{2n}; f5 = 100^{2n}; f6 = n^4(logn)^2$
Since exponentials always grow faster than polynomials, in the order of running time complexity higher to lower:
$100^2n > 10^2n > n^5 > n^4(logn)^2 > (n + 2)^2 > n2n$
Thus:
$f5 > f4 > f1 > f6 > f3 > f2$

(3: Ch#2 Ex#4)

$g1 = 2^{sqrtlogn}; g2 = 2^n; g3 = n(logn)^3; g4 = n^{\frac{4}{3}}; g5 = n^{logn}; g6 = 2^{2^n}; g7 = 2^{n^2}$
Since exponentials grow faster than polynomials: $g1, g2, g6, g7$ are definitely asymptotically larger than the rest.
$2^{2^n} > 2^{n^2} > 2^n > 2^{\sqrt{logn}}$
Considering $g3, g4, g5$:

$n^{logn} > n^{\frac{4}{3}} > n(logn)^3$
Thus:

$g6 > g7 > g2 > g1 > g5 > g4 > g3$

(4: Ch#3, Ex#5)

Given: $f(n) = O(g(n)) \implies f(n) \leq cg(n) for\ c > 0, \forall n \neq n_0$
Part (a): $log_2(f(n))\ is\ O(log_2(g(n)))$
$f(n) \leq cg(n) for\ c > 0, \forall n \neq n_0$
As $g(n)$ is positiev defnite, a lesser strict bound on f(n) is given by:

Taking logarithm on both sides: $log_2(f(n)) \leq log_2(g(n)) + log_2(c)$ for $c > 0, \forall n > n_0$
if $|g(n)| > 2$ the RHS would be positive definite, and $log_2(f(n)) = O(log_2(g(n)))$ would be true. But this is not true in general($|g(n)| < 2$)
$log_2(f(n)) \leq log_2(g(n)^d) \implies log_2(f(n)) \leq dlog_2(g(n))$, for some $d > 1, \forall n > n_0$
Which clearly proves Part (a), but is a special case and hence **(a) is FALSE in general.**
Part (b) $2^{f(n)}\ is\ O(2^{g(n)})$
Consider $f(n) = 2n^2$, and $g(n) = n^2$, then $2^{f(n)} = 2^{2n^2}$ while $2^{g(n)} = 2^{n^2}$, which clearly does not satisfy the given relation.
Hence FALSE.
Part (c)
Since, $f(n) \leq cg(n) for\ c > 0, \forall n \neq n_0$, squaring both sides:
$f(n)^2 \leq c^2 g(n)^2 for\ c > 0, \forall n \neq n_0 \implies f(n)^2 \leq c' g(n)^2 for\ c' > 0, \forall n \neq n_0$
Thus Part(C) is TRUE.

(5: Ch#2, Ex#6)

Part (a, b): There is one loop involved besides the for i,j loop that calculates the sum of A[i] through A[j], which is equal to $\sum_i \sum_j (j-i) = \sum_i^n \sum_{j=i+1} n(j-i) = \sum_i^n n(n+1)/2 = O(n^3)$
Part (c): To avoid summing A[i] through A[j], we rely on $B[i,j] = B[i,j-1] + A[j]$ initialising B[1,1] as A[1].

(6: Ch#3, Ex#4)

We assume the given statement to be true. Consider n nodes with two children each and m leaves. The Binary tree is not necessarily balanced. $n - m \leq 1$
Adding one more node to this tree will lead to $n+1$ nodes. Now there are two scenarios:
Case1 : The binary tree was balanced and adding one node lead to an imbalanced tree.
This will cause the number of leaves to remain same,'m' as the new node is entirely at a new level(depth), thus accounting for one leave and the nodes at the penultimate level are all leaves except the one whose child is this new node. So we have $m' = 1 + m - 1 = m$ leaves and $n' = n$ nodes with two children. Since $n - m \leq 1 \implies n' - m' \leq 1$. Adding a new leaf to a balanced binary tree does not lead to either more leaves or more nodes with two children.(the added leaf's parent has a single child!)
Case2 The binary tree was imbalanced so the new node is added to the same level as the leaves were. Thus the number of leaves go up by 1. The node to which this is added may transform into a parent with two or one child and so $n' = n \ OR \ n+1$ depending on where the new leaf is added. and $m' = m+1$ in either cases. Since $n - m \leq 1 \implies (n - (m+1) \leq 0; (n + 1 - (m+1) \leq 1)$, Thus $n' - m' \leq 1$
Hence Proved.

(7: Ch#3, Ex#6)

Let $(x,y) \in G$ be an edge such that $(x,y) \notin T$, G rooted at u. Since (x,y) are connected they must occur in some layer of T. Since T is a DFS tree, x,y must be ancestor of each other. Say x is an ancesotr of y. At the same time T is a BFS tree, and hence the difference of distance of (s,x) and (s,y) can differ at maximum by 1 and they are ancestors, so they should be connected in T via an edge too.

(8)

Let the depth of DFS tree T be less than that of BFS tree U. Now consider the leaf of T can then be reached from v in a shorter distance than in U since the depth of T is smaller, which is a contradiction to BFS tree property of providing the shortest distances.