CSCE 500 Midterm Exam #1

9/30/2015

1. Solve the two recurrences below:

   $T(n) = 2 \cdot T(n/4) + n^{1/2}$ (7%)

   $T(n) = 2 \cdot T(n^{1/2}) + \lg_2 n$ (10%)

2. Show the following non-asymptotically tight bound:

   $n^2/\lg_2 n = o(n^2)$ (3%)

3. Prove that any comparison-based sort algorithm over n keys requires the complexity of $\Omega(n \cdot \lg_2 n)$, by making use of the Stirling's approximation below: (12%)

   $$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \left(1 + \Theta\left(\frac{1}{n}\right)\right)$$

4. HEAPAORT starts with building a binary heap through the procedure below:

   MAX-HEAPIFY(A, i, n)

   l = LEFT(i)

   r = RIGHT(i)

   if $l \le n$ and $A[l] > A[i]$

   1.

   2.

   if $r \le n$ and $A[r] > A[largest]$

   3.

   if $largest \ne i$

   4.

   MAX-HEAPIFY$(A, largest, n)$

   Max SubArray Algo

   (a) Complete the missing four statements (indicated). (10%)

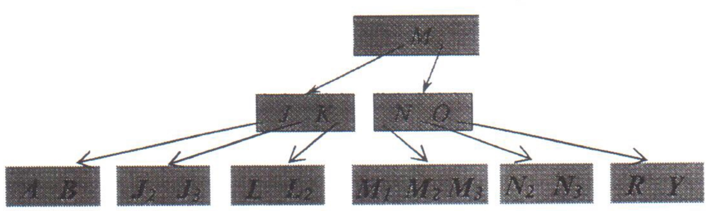
   (b) What is the time complexity upper bound expression of the MAX-HEAPIFY procedure, given that a binary heap is based on a complete binary tree? (5%)

5. QUICKSORT(A, p, r) relies on PARTITION(A, p, r) to fragment Array A into two subarrays based on $A[r]$ (called the pivot). Given Array $A = \{11, 8, 25, 14, 16\}$, follow QUICKSORT to sort the array elements by showing the result of every PARTITION(A, p, r) call, i.e., its fragmented two subarrays at the end of that call. (10%)

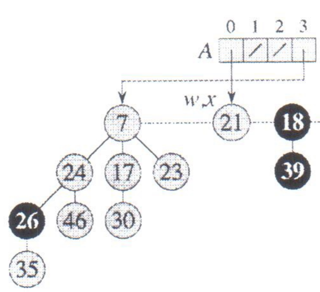
6. The binary search tree (T) facilitates key search and it involves several operations to maintain the tree property when a node (z) is deleted, as shown in the following pseudo code, TREE-DELETE(T, z), where TRANSPLANT(T, u, v) replaces the subtree rooted at u with one rooted at v. Fill in the last three missing statements in the pseudo code below. (10%)

```
TREE-DELETE(T, z)
if z.left == NIL
    TRANSPLANT(T, z, z.right)          // z has no left child
elseif z.right == NIL
    TRANSPLANT(T, z, z.left)           // z has just a left child
else // z has two children.
    y = TREE-MINIMUM(z.right)          // y is z's successor
    if y.p ≠ z
        // y lies within z's right subtree but is not the root of this subtree.
        TRANSPLANT(T, y, y.right)
        y.right = z.right
        y.right.p = y
    // Replace z by y. Transplant (T, z, y)
    [ y.left = z.left ]
    [ y.left.p = y ]
    [              ]
```

7. Given the initial B-tree with the minimum node degree of $t = 3$ below, show the results (a) after deleting two keys in order: M then R and (b) followed by inserting the key of L_l, with $L < L_l < L_2$. (Show the result after each deletion and after insertion; 10%)



8. A Fibonacci min-heap relies on the procedure of CONSOLIDATE to merge trees in the root list upon the operation of extracting the minimum node. Given the following partially consolidated diagram, show every subsequent consolidation step till its completion. (10%)



Good Luck!

nishyanand@gmail.com