Indian Statistical Institute, Bangalore

CS3, End Semester Examination, November 2013 Max Marks:50; Weightage 50%, Max Time: 3 hrs

NB. Unless otherwise mentioned assume graphs are simple connected.

1. An engineer plans to build a radix sort algorithm. $Sort(k,A)$ is a method available to sort elements of array A based on the k^{th} bit value of the numbers in A. *(3+2+2=7)*
    a. Write out the radix sort algorithm using the available sort method.
    b. What must be the property that method must satisfy, beyond being able to sort.
    c. What is the complexity of the resultant algorithm?
2. G is a weighted simple connected graph with non-zero edge weights. You wish to find the spanning tree by Prim's method which grows the spanning tree from a given vertex. Present Prim's algorithm. If G is represented by an adjacency matrix what is the complexity of the algorithm? *(4+1=5)*
3. Given a connected DAG D: *(3+1=4)*
    a. Sketch an efficient algorithm find its topological sort, with a root appearing as the first item in the sorted order.
    b. What is the complexity of your algorithm if you use an adjacency list to represent D?
4. Provide a counter example: If u and v are articulation points of a simple graph G then $G - (u,v)$ is disconnected. *(2)*
5. You are given an array $A[1..n]$ of structures with a certain field called "key". You wish to create an array $B[1..n]$ which contains indices to A so that if $i < j$ then $A[B[i]].key < A[B[j]].key$. Mention how this can be done using a standard sort algorithm in $O(n \log(n))$ time. *(4)*
6. T is a DFS tree of G. Let $d(x)$ for any node x represents its distance from the root in T. For an edge $e=(u,v)$ of G not in T mention ***True*** or ***False*** for each of the following statements: *(2+2+2+2=8)*
    a. e does not connect a node to its ancestor in T
    b. $d(u) < d(v)$ implies $pre(u) < pre(v)$
    c. $|d(u) - d(v)| > 1$
    d. Distance from u to v in $G-e$ is $d(u)+d(v)$
7. The complexity of quicksort(worst case) is $\Theta(n^2)$. Assume we write a partition algorithm that takes $\Theta(n)$ time for partitioning an n element array and also guarantees at least $1/4^{th}$ of the elements in each of the partitions. Analyze the worst case complexity of quicksort that uses the modified partition method. *(4)*
8. We wish to implement a database. We store the keys in a 2-3 tree. If the height of the tree is h, what is the maximum number of comparisons to decide if a key exists in the database? *(2)*
9. In class we studied a binary heap on an array. In this question you will create a min-heap on an array based on a 3-ary tree: *(2+2+2+2=8)*
    a. Assuming indices start at 1, for an element at index i, what are the indices of its parent and its three children.
    b. Compute the height h, of such a heap on n elements.

c. Given a 3-ary heap with n elements, give the algorithm to insert a new element. Analyze the complexity.
d. Given a 3-ary heap with n elements, give the algorithm to remove the minimum element. Analyze the complexity.

10. You have a basket that holds at most weight W of material. There are n articles available each with a certain weight w_i and a certain value v_i. Your aim is to fill the basket with materials that add up to give the maximum value, without exceeding the total weight W. You employ the following simplistic heuristic: Pick the items in order of decreasing value until the basket allows no more. *(1+5=6)*
    a. Show with a simple example with three articles that this method does not yield the best choice.
    b. Sketch an algorithm that implements the above method in $\Theta(k \log(n))$ time where k is the number of articles that you end up picking