Written Theory Qualifier Exam

Your number: ________________________________

Time limit: 2.5 hours. Use only the notes supplied by the Department

SPRING 2013, CS DEPARTMENT, IIT

For every question, please write your answer in a clean and concise way. Use additional pages, start a new page with each problem and write only on one side of the paper.

Use procedures if you want – – marking clearly what the parameters are and what they do, and with what running time in terms of its parameters. Unless the procedures are from the textbooks, write pseudocode for the procedures.

Problem 1. The SELECTION algorithm from Subsection 9.3 uses partitioning into groups of 5, computing the median in each group, followed by two recursive calls (with some work between them). Analyze the running time for two variants of this algorithm, where the groups are of size 3, and of size 7 respectively. Theta-bounds are needed for full marks.

Problem 2. Let T be a max-heap storing n keys. Assume for this problem that the heap is stored in an array A starting from 1 - that is, $A[1]$ is the biggest key in the heap. Give an efficient algorithm for reporting all the keys in T that are greater than or equal to a given query key x (which is not necessarily in T). Note that the keys do not need to be reported (printed) in sorted order. Analyse the running time. An $O(k)$ algorithm is needed for full grade, where k is the number of elements printed. That is, the algorithm should do a constant number of elementary operations per printed element.

Present pseudocode first.

Problem 3.

We describe below a data structure that maintains the transitive closure of a directed graph while arcs (directed edges) are added to the graph.

Formally, a set of vertices V is given (with $|V| = n$), and arcs $e_1, e_2, \ldots, e_m$ become available one by one (e_i is not known before computing R_{i-1}, defined below). Let $G_i = (V, E_i)$, where $E_0 = \emptyset$ and $E_i = E_{i-1} \cup e_i$. Let R_i, a $n \times n$ matrix, have $R_i[u, v] = 1$ if u has a directed path to v, and $R_i[u, v] = 0$ otherwise. Thus R_i stores the transitive closure of G_i.

Note that R_0 has entries that are 1 only on diagonal.

1. Give a series of instances (one for each n) such that there exists an i with the number of entries 1's in R_i being $\Omega(n^2)$ higher than the number of entries 1 in R_{i-1}.

2. Consider however the code

   ADD(e_i) where the tail of e_i is u and the head of e_i is v:

   ```
   for all x ∈ V
     if R[x, u] = 1 AND R[x, v] = 0
       for all y ∈ V
         R[x, y] ← max(R[x, y], R[v, y])
   ```

   Prove that if $R = R_{i-1}$ before the code is executed, then $R = R_i$ after the code is executed.

3. Use the first part of this problem to show that ADD(e_i) may have running time $\Omega(n^2)$.

4. Prove that despite this, the running time of m operations ADD(.) is $O(nm + n^3)$.