Design and Analysis of Algorithms

1 Semester, 2024-25, Indian Statistical Institute, Bangalore

Mid term Examination (Compensatory); marks 60, Time Limit - 2 hours

October -, 2024

1. Answer briefly

[15]

(a) A program sorts an array of n elements followed by search k times on the sorted array. What is the overall **running time** (in Θ notation) of the program assuming efficient algorithms for both sorting a searching. What are the algorithms do you assume for the sorting and searching. (no need to explain) [3]

- (b) The simple algorithm to find if a number n is prime (i.e., by testing divisibility of n by numbers from 2..n 1) is considered an exponential time algorithm. Why? [2]
- (c) What are the complexities of the following two operations on efficient implementations of a binary search tree and a binary max-heap: (i) Search for a key and (ii) decrease a key (no need to explain) [4]
- (d) Write the pseudo-code for the BFS algorithm on a graph G(V, E). Mention any data structures you use. What is the complexity of the algorithm? [6]

 (a) Write the pseudo-code for an efficient implementation of Kruskal's algorithm for a min weight spanning tree. Mention which representation for the graph you are using (adjacency matrix or adjacency list) and any data structures you use. Derive the complexity. [12]

(b) Show that it is possible to have an even more efficient implementation if the edges had weights in the range 1..|V| where |V| is the number of vertices in the graph. [3]

- 3. Answer assuming a binary max heap on an array with indices 1..n.
 - (a) What is the order relationship between values in the array when it is a heap? [2]

(b) Write pseudo-code for a linear time algorithm to convert an arbitrary array into a binary heap. [9]

(c) Do the analysis to show that it is indeed linear time. You may assume the summation for well known series without deriving them. [4]

4. (a) Write out the pseudo-code for the standard algorithm for DFS on a directed

graph D, giving every node a pre and a post number.

[4]

- (b) Mention how you would use the above to detect cycles in the directed graph (what would you do at which step?). [3]
- (c) If the digraph was a DAG, mention how you would use the above to create a topological sort of the nodes in the graph. [3]
- (d) Run your DFS algorithm on the following DAG clearly showing pre and post numbers for all nodes and finally the corresponding topologically sorted order of the nodes. Mark the pre-numbers with a circle around the number and post number with a square around the number. Clearly mark the nodes that DFS is initiated at and the edges which are part of the DFS tree. [5]

