

Q1.

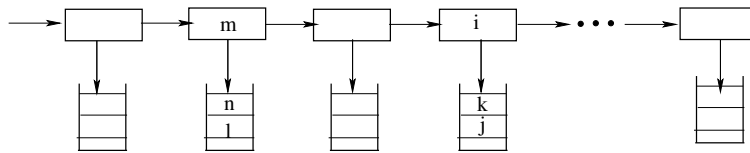
(a) You are given an array of integers A of length n and another integer q . You have to design an $O(n \log n)$ algorithm to either find a pair of elements of A such that their sum is q , or report that no such pair exists.

(5)

(b) Prove that the time complexity of the algorithm you have designed in part (a) is $O(n \log n)$. Prove the correctness of your algorithm. Intuitive arguments will suffice for both.

(5)

Q2. You have to design a *list of stacks* data structure. It is a linked implementation of a list at the first level. A stack is associated with each node of the linked list at the second level. Every input to this data structure is a pair of integers (i, j) . For inserting an item (i, j) you need to look for a node in the linked list that stores the item i , and push the item j on the stack associated with that node. In case you cannot find a node of the linked list storing item i , you have to insert a new node in the linked list and store item i in that node and item j in the associated stack. Deletion of a pair (i, j) is similar, the item j has to be deleted from the stack of the node of the linked list that stores item i , or an error should be reported if no node of the linked list stores item i . (See the picture below)



insertion of (i,j) , (i,k) , (m,l) , (m,n) in the list of stacks data structure

(a) Write Java classes for implementing the *list of stacks* data structure. Write constructors for each class that you use. You should write correct Java code for this question and explain the code clearly in English.

(5)

(b) Write `insertion` and `deletion` methods for the *list of stack* data structure. These need not be correct Java code, but should be clearly written if you write in English.

(5)

Q3.

(a)

- Prove that $n^3 \log^{10} n$ is $O(n^4)$.
- Prove that $n^2 \log n$ is not $O(n^2)$.

(5)

(b) Write pseudo code or explain in your own words Prim's minimum spanning tree algorithm. What is the time complexity of this algorithm?

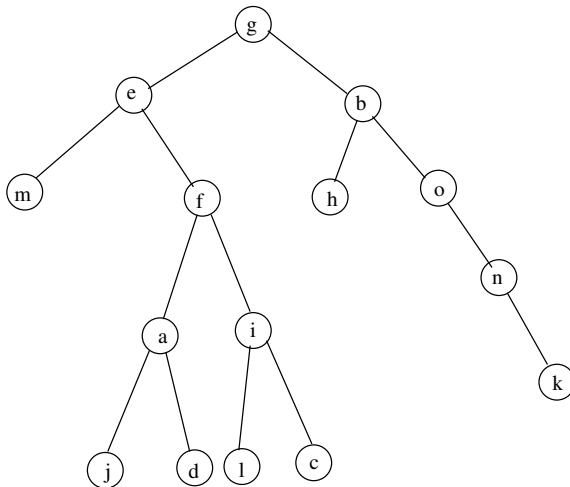
(5)

Q4.

(a) Write pseudo code for the recursive and non-recursive depth-first search algorithms. What are the time complexities of these two algorithms?

(5)

(b) Show the execution of your non-recursive depth-first search algorithm in part (a) on the tree given below. Show the status of the stack at every step of depth-first search.



(5)

Q5.

(a) Write pseudocode or explain in your own words Dijkstra's single-source shortest-path algorithm. Given a graph $G = (V, E)$, what is the asymptotic complexity of this algorithm? Explain your answer clearly. (5)

(b) Write pseudo code or explain in your own words the matrix-multiplication based all-pairs shortest path algorithm. Given a graph $G = (V, E)$, what is the asymptotic complexity of this algorithm? Explain your answer clearly. (5)

—————END OF PAPER—————