ESO207: Data Structures and Algorithms

Theory Assignment 2

Due Date: 22nd March, 2021

Total Number of Pages: 4 Total Points 100

Instructions

1. The assignment contains 2 parts - Part 1 and Part 2. Please submit both parts in separate files titled `part1_roll.pdf` and `part2_roll.pdf` respectively (where `roll` is your roll no). Failure to do so will result in loss of marks.
2. All solutions must be typed on a word processing application such as LaTeX or Word. Handwritten solution will not be accepted.
3. For each question you must give the pseudocode of your algorithm and a brief description of the idea of your algorithm.
4. If an algorithm requires a certain time complexity or space complexity, then you must describe why your algorithm indeed works in that complexity bound.
5. The teaching assistant in charge of Part 1 is Koustav Bhanja (`kbhanja@cse.iitk.ac.in`) and in charge of Part 2 is Ashish Ranjan Yadav (`aryadav@cse.iitk.ac.in`). Contact them if you have any doubts.

Part 1

Problem1. (20 points) Let $G = (V, E)$ be any unweighted directed graph on $|V| = n$ vertices and $|E| = m$ edges with source $s \in V$ and destination $t \in V$. Moreover t is reachable from s.

G is said to be *1-connected* if there exists a subset $E_1 \subseteq E$ such that if we remove any one edge $e \in E_1$ then it partitions the vertex set V into two non empty subsets S, T such that there is no path from $s \in S$ to $t \in T$ in $G = (V, E \setminus \{e\})$.

Consider G to be a given *1-connected* unweighted directed graph. Preprocess G in $O(n + m)$ time and build a data structure of size $O(n + m)$ to answer the following 1-*connectivity* query in $O(1)$ time,

$$\text{1-connectivity}(G, e) = \begin{cases} 1 & \text{if graph } G \text{ is still } \textit{1-connected} \text{ after removing } e \in E \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

Give pseudo code for preprocessing step and prove the time and space complexity. Also give the pseudocode and explain how a query, *1-connectivity*(G, e) is handled using the constructed data structure in $O(1)$ time.

Problem2. (10 points) An array A is given with n numbers. A pair (i, j) with $0 \leq i < j < n$ is said to be *strongly dominated* if $A[i] > 2A[j]$. Design an $O(n \log n)$ time algorithm that reports the total number of *strongly dominated* pairs in A and prove the time complexity of your algorithm.

Problem3. (20 points) Let $P = \{p_1, p_2, ..., p_n\}$ be a set of n points given in $2D$ where $p_i = (x, y_i)$ and $y_i \neq y_j$ for all $i \neq j$. Additionally two more points $s = (x_s, y_s)$ and $t = (x_t, y_t)$ are given where $x_s < x$ and $x_t > x$. A path between s and t must contain exactly one point $p_i \in P$, (In other words, it is a path of length 2, $s \mapsto p_i \mapsto t$, where $p_i \in P$). Cost of an edge is the Euclidean distance between the pair of points. Hence the cost of a path is the sum of distances of every edge in the path. Your task is to build a compact data structure in $O(n \log n)$ time and $O(n)$ size that will be able to answer the following *Min-Cost-Point* query in $O(\log n)$ worst case time.

Min-Cost-Point$(P, s, t) = \{\, p_k \mid p_k \in P$ and the cost of path between s and t including p_k is minimized.$\}$.

Query outputs an appropriate point p_k from P such that the cost of the path between s and t including point p_k is minimized, for any given s and t.

Also you are not allowed to sort the points in P. Assume that computing distance between a pair of points takes $O(1)$ time, further computing the cost of a 2-path takes time $O(1)$.

Give the pseudo code for preprocessing step and prove the time and space complexity. You must also give pseudo code or explain how a query is handled using the constructed data structure in $O(\log n)$ time.

Part 2

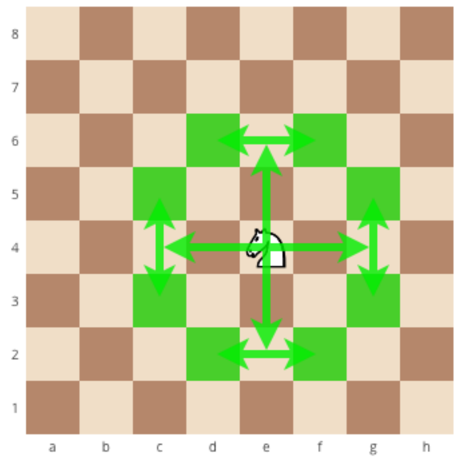
Problem4. (15 points) Given an array A of integers , you have to return an array *count* where each index i, the array $count[i]$ is the number of smaller elements to the right of $A[i]$.

Give $O(n \log n)$ runtime pseudo code to solve the above problem. Prove the time complexity of your algorithm.

Example :
Input: $A = [5, 2, 6, 1]$
Output: $[2, 1, 1, 0]$
Explanation:
To the right of 5 there are 2 smaller elements (2 and 1).
To the right of 2 there is only 1 smaller element (1).
To the right of 6 there is 1 smaller element (1).
To the right of 1 there is 0 smaller element.

Problem5. (15 points) Give an algorithm that determines whether or not a given undirected graph $G = (V, E)$ contains a cycle. Your algorithm should run in $O(V)$ time, independent of the number of edges in G. Mention the data structures used in the algorithm. Analyze the time complexity of your algorithm.

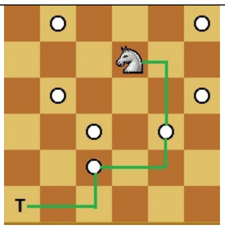
Problem6. (20 points) A knight is a chess piece that moves in an L shape i.e the knight moves 2 units in one direction (i.e., horizontal or vertical) and 1 unit in the perpendicular direction. The figure below shows all the possible moves of knight from position $(e, 4)$.



You are given a square chess board of size $N \times N$. The knight is initially at source point, (C, D) and destination point of the knight is (E, F). You need to find the minimum number of steps for the knight to move from source to destination. Note that the knight should not move out of the chess board at any point in time.

Give the pseudocode along with the analysis of time complexity and space complexity for the algorithm.

Example :



In the above example the knight takes 3 step to reach the target T i.e from (1,4) to (6,1).
$(1, 4)- > (4, 5)- > (5, 3)- > (6, 1)$.