ESO207A: Data Structures and Algorithms

Theoretical Assignment 3

Due Date: 3rd November, 2023

Total Number of Pages: 4 Total Points 100

Instructions-

1. For submission typeset the solution to each problem and compile them in a single pdf file. Hand-written solutions will not be accepted. Use LaTeX or similar word processing application for typesetting.
2. Start each problem from a new page. Write down your Name, Roll number and problem number clearly for each problem.
3. For each question, give the pseudo-code of the algorithm with a clear description of the algorithm. Unclear description will receive less marks. Less optimal solutions will receive only partial marks.
4. Don't add any screenshots of code, etc. in your solution.
5. Please refrain from attempting problems without proper consideration. In case of uncertainty, clearly indicate "I Don't Know", 10% of the total marks will be awarded for such responses.
6. Do not add unnecessary adjectives and very general comments/terms in your answer.

Question 1. All or None (15 points)

Picture a playful adventure in a land of cities and roads! Meet Comren, a curious explorer ready to roam this exciting world. The cities are dots on a map, and the roads are the lines connecting them. This can be represented as an undirected graph representing a network of cities connected by roads. Comren won't settle for less, it's no fun to repeat or miss any road, all or none. Your job? Travel to every city once, using each road exactly once, and finally return to where you started, if at all possible.

(a) (2 points) Given a Depth-First Search (DFS) traversal, can you find such a path? Why or why not?

Solution: No, DFS can be used to find a path that visits all cities, but it may not guarantee a path that traverses each road exactly once unless certain conditions are met (see part c).

(b) (2 points) Given a Breadth-First Search (BFS) traversal, can you find such a path? Why or why not?

Solution: Similar to DFS, BFS can also be used to find a path that visits all cities. However, BFS also may not guarantee a path that traverses each road exactly once. These algorithm is typically used for problems that require traversing each vertex (not edge) exactly once.

(c) (4 points) What conditions should the graph meet for you to be able to traverse each road exactly once and return to the starting city?

Solution: Such a path is referred to as the Eulerian Cycle or an Eulerian circuit. An undirected graph has an Eulerian cycle if the following two conditions are true:

1. All vertices with non-zero degree are connected. We don't care about vertices with zero degree because they don't belong to Eulerian Cycle or Path (we only consider all edges).
2. All vertices have even degree.

(d) (7 points) If the graph meets the conditions mentioned in part c, outline the steps you would take to find such a path. (Best Complexity: $O(|E|)$, 3 points for $O(|E|^2)$)

Solution: The standard way to get to get Eularian cycle is via Hierholzer's Algorithm. In Hierholzer Algorithm, we can follow the steps below to find the Eulerian Cycle in a graph:

1. We can choose any arbitrary vertex as a starting point
2. We follow the edges of the vertex that we haven't followed before and traverse the vertices (Initially, all the edges are unfollowed of course). We can follow whichever edge we want. The only rule is not to follow previously traversed ones.
3. We should apply Step 2 until we are stuck. At some point, we will visit a vertex, and there will be no edges to follow. Remember that Eulerian Cycle properties, every vertex should have even degrees. If we are stuck the first time, it means that we formed a cycle, and the vertex that we are stuck on is the starting vertex. we returned where we started.
4. Push the vertex that we stuck to the top of the stack data structure, which holds the Eulerian Cycle.
5. Backtrack from this vertex to the previous one.
6. If there are edges to follow, we have to return to Step 2.
7. If there are no vertices left to traverse, now the stack holds the complete Eulerian Cycle, and we are done. Otherwise, we have to follow Step 5.

Question 2. Chaotic Dino (25 points)

We have chaotic Dinosaurs dormant(which can be activated by a signal) across all cities in a state. The state has several cities connected via roads, which can be represented as an undirected graph.

Some of these cities have towers. Each tower has the same power, say x. Towers can send signal to cities whose shortest distance from the tower is at most x. A tower is activated iff the city where the tower is situated receives a signal from another tower.

If the city receives a signal, the dormant dinosaurs destroy the city. You, an evil mastermind, have a list of all cities with a tower and a map of the state. Your source city is S. You want to destroy the destination city D.

Note: Source city S has a tower, and you can only activate this tower. However, you can also configure the power of every tower, x.

(a) (15 points) Write clear pseudo code and explain logic briefly to check whether the power of x for all towers will be able to send a signal from S to D.

Solution: Do a BFS from the S node till a depth of x, and repeat the BFS for each tower city encountered till you reach Node D or run out of nodes.
Complexity: $O(V(V+E))$

(b) (10 points) Write clear pseudo code and explain logic briefly to obtain the minimum power of tower required to send a signal from S to D.

Solution: Do a binary search on range $(0, V]$ against the logic implemented in part (a) to get the minimum possible value of x permissible.
Complexity: $O(V(V+E)\log(V))$

Question 3. Room Colors (30 points)

Shantanu and his friends are very excited to visit their home during the midsem break, being the rowdy bunch that they are they decided to color bomb their hostel before leaving. Upon leaving he realizes that he no longer knows what color the rooms are, and since he is the hall president it is very important to know what the final color of every room is, Shantanu must figure out the colors on his way home, he remembers who fired what shots and in what order, but since he lives in a big hostel with a lot of rooms, he is unable to perform the requisite calculations on his own. Luckily he finds you as his co-passenger on the ride home.

There are n rooms in the hostel, you are given m bombings in chronological order and each bombing is of the form (l, r, c) which means all rooms from l to r where $1 \leq l < r \leq n$ were bombed with color c. Give an $O((m+n)\log n)$ time algorithm to help him find the final color of every room.

Solution: Proceed with the queries in reverse chronological order, maintain a dsu where each cell's parent is the leftmost unpainted cell to its right. Note that since we are going in reverse chronological order, each room needs to painted only once as that would be its final color.

Question 4. Fest Fever (10 points)

While he is home on vacation, Shantanu takes his little brother, Anuj, to visit the ongoing festival fair, his brother insists on eating from some consecutive set of sweets from a stall, Shantanu was aware something like this would happen and had therefore asked the prices of all the sweets beforehand. The prices may change as the fest progresses but being the Bacchan of Shastri Nagar, he'll get to know as soon as a price changes.

Shantanu has some money M, when his brother makes a request of the form (l, r) where $1 \leq l < r \leq n$, he checks if he has enough money, if so he fulfills his request, also, note that as soon as a request is fulfilled, Shantanu is returned all his money since the vendor doesn't want to incur his wrath.

Given the money M an n queries (updates + requests) in chronological order, give an $O(n \log n)$ time algorithm to that outputs "YES" if a request can be fulfilled and "NO" if it cannot.

Solution: Mantain a segment tree.

Question 5. Edible sequence (20 points)

It is time for Shantanu to return home after the midsem break, since he was a little sad leaving home, his mother packed fresh home grown apples for the ride back to campus.

Shantanu, being the nerd he is has numbered all the apples growing on his mother's apple tree and made a tree structure using the apples as nodes. He finds you on his ride back and tells you all about his tree. Now you're both getting hungry but the sequence of apples is edible iff you eat them in a valid BFS order.

Given the tree T with n apples, and a sequence of n apples, give an $O(n)$ time algorithm to help Shantanu find if the sequence is edible or not.

Solution: Do a BFS on the tree, the sequence given is only valid if the depths are in non-decreasing order and the children of the same parent occur consecutively.