CSE200A: Competitive Programming I (Summer 2019) Reading 3

Apoorv Singh, Yatin Arora, Prakhar Gupta

July 2019

1 Minimum Spanning Tree and Disjoint Set Union

A Minimum Spanning Tree is defined with respect to a weighted graph. It comprises of all the vertices of the graph and includes a subset of edges present in the graph such that the new graph is a tree and the total weight is minimum.
A disjoint-set data structure (also called a union–find data structure or merge–find set) is a data structure that tracks a set of elements partitioned into a number of disjoint (non-overlapping) subsets. It provides near-constant-time operations (bounded by the inverse Ackermann function) to add new sets, to merge existing sets, and to determine whether elements are in the same set. Apart from kruskal's, it plays a key role in a plethora of other problems.
Refer to the **Section 6.1 - Minimum Spanning Trees** from the textbook "The Algorithm Design Manual by Steven S. Skiena". You can also read more on Disjoint Set Union here.

1.1 Problems

- https://www.codechef.com/problems/ABROADS
- https://www.hackerearth.com/practice/algorithms/graphs/breadth-first-search/practice-problems/algorithm/containers-of-choclates-1/
- https://www.hackerearth.com/challenges/competitive/code-monk-graph-theory-i/algorithm/monks-birthday-treat/
- https://www.codechef.com/SEPT17/problems/FILLMTR

2 Trie Data Structure

Trie is a data structure generally used for storing strings with a provision of fast search. The time complexity for searching a string stored in a trie is O(L) where L is the length of the

string to be searched. But tries can be used to solve a wide variety of problems not just limited to fast string search.

2.1 Resources

- https://www.topcoder.com/community/competitive-programming/tutorials/using-tries/
- https://threads-iiith.quora.com/Tutorial-on-Trie-and-example-problems - Try to code all three problems explained in this blog.
- https://www.hackerearth.com/practice/data-structures/advanced-data-structures/trie-keyword-tree/tutorial/

2.2 Problems

- Try solving problems mentioned in the resources.
- https://www.codechef.com/JUNE18B/problems/SHKSTR

3 Shortest Path Algorithms

The shortest path problem is about finding a path between vertices in a graph such that the total sum of the edges weights is minimum.
This problem could be solved easily using (BFS) if all edge weights were 1, but here weights can take any value.

3.1 Resources

- **Dijkstra:** HackerEarth, Visualgo
- **Shortest Path in DAG:** GFG
- **Bellman Ford Algorithm:** Youtube, Cp-Algo
- **Floyd–Warshall's Algorithm:** Youtube

3.2 Problems

- **Easy**
- Shortest-Path
- HackerEarth
- DIGJUMP
- **Hard**
- **Bonus**

4 Segment Tree

You need to study only the basics i.e. only the following functionalities:

- Range querying (eg. what is the minimum in the array between index 12 to 43).
- Point updates (eg. multiply the array element at 15th index by 7).

4.1 Reading

Read the following:

- Algosaurus - Read everything before lazy propagation.
- Codeforces - Read everything before lazy propagation.
- Hackerearth

4.2 Practice Problems

- Range Minimum Query - HackerRank
- 380C - Codeforces
- 482B - Codeforces (Try to do this problem using segment tree)