CS210A: Data Structure and Algorithms

Semester I, 2014-15, CSE, IIT Kanpur

Programming Assignment 5

Graphs

Note: There are 2 problems in this assignment. The first problem carries 30 marks and the second problem carries 70 marks. The input and output formats of these problems will be specified on the judge by 8th November.

Problem 1

In this course and the course on Discrete Mathematics, you would have studied the fascinating concept of Graphs, their structure and their properties. However, you may not have yet appreciated the usefulness of graphs in the field of Computer Science and beyond.

Graphs are very useful when one needs to model pairwise relations between objects, and such a setting is encountered in variety of situations. Various real-world and difficult problems have been solved by modeling the problem (and hence reducing) to a graph, and then using the familiar (and well-researched) concepts of graph theory to infer results about the original problem. The use of graphs has not just been limited in Mathematics and Computer Science; numerous scenarios, from optimizing the network topology, to discovering the anatomy of the brain, to finding an efficient road or water-transfer network, all have been efficiently modeled and studied as a graph.

Often, the difficulty (and the novelty) in reducing a problem in such a way is finding an efficient graph model for the problem. The aim of this exercise is to introduce you to such a problem (which initially might seem unrelated to graphs) and encourage you to solve it using your knowledge of graph theory.

The problem is as follows :
A $perfect - square$ is an integer that is the square of an integer.
You can move from one $perfect - square$ to another $perfect - square$ if they have equal number of digits and the number of places where the two $perfect - square$ have different digits is **exactly two**. Any other kind of move between two numbers is not allowed.
For example, 10000 → 10201 is a valid move, but 10000 → 90000 is not valid (since they differ at 1 position only). Similarly, 10000 → 10816 or 10000 → 11236 are not valid (differ at 3 and 4 places respectively). Also 10000 → 10810 is not valid, since 10810 is not a $perfect - square$.

You are given two **five-digit** *perfect-squares*, a and b. You have the find the minimum number of moves required to reach b from a. If no such set of moves exist, output -1.

For example, suppose $a = 10000$ and $b = 95481$, then the configuration of minimum moves required would be :
10000 → 10201 → 22201 → 25281 → 95481
Hence the minimum number of moves required is 4.

Problem 2

Recall Lecture 36 in which we discussed two algorithms for MST. One algorithm was based on cut property and we gave an $O(mn)$ time implementation in the class. You have to give an $O(n^2)$ time implementation of this algorithm. However, you must follow the two constraints given below otherwise you won't get any credit for this assignment:

1. You may use either Adjacency lists or Adjacency matrix for this problem.
2. You must <u>not</u> use any data structure like Binary heap, Binary search tree.
3. You have to design an $O(n^2)$ time *implementation* of the algorithm that we discussed in the class. So you are <u>not</u> allowed to design any new algorithm.