ESO207: Data Structures and Algorithms

Programming Assignment *5* *Due: 16 November 23:59 hrs*

Problem 1. Graphs: Simple BFS/DFS. You are given a directed graph $G = (V, E)$ represented by an adjacency list. Two vertices are specially marked, *source* vertex and *target* vertex. All other vertices are marked as *donors* or *robbers*. A traveller starts at the source vertex and has to find his way to the target vertex. On the way, if the traveller arrives at a donor vertex, then he gets some gifts and money. If the traveller arrives at a robber vertex, he gets robbed completely of his belongings and is imprisoned. You have to design an algorithm that returns some path from the source vertex to the target vertex that avoids robber vertices, if such a path exists, and otherwise, returns *False*.

The input will be given in the following format. The first line specifies the number of vertices n, the source vertex number and target vertex number. The next n lines each, except for the lines corresponding to the source and target vertices, starts with a character R or D (for robber or donor) followed by the adjacency list of vertex i in some order (i.e., the list of vertices to which vertex i has an outgoing edge). The lines corresponding to the source and target vertices only contain the adjacency list, since they are neither robbers nor donors.

The output should be in two lines. The first line should say *True* if there is a successful path from the source to the target without encountering a robber vertex or it should say *False*. In the case of *True* output, the path from the source vertex to the target vertex (including both of them) should be printed. A path is a sequence, for example, 2 4 3 5 would be the path consisting of the sequence of edges $(2, 4), (4, 3), (3, 5)$.

Problem 1. Bonus Problem: Shortest Paths This is a generalization of the above problem. There are now two kinds of vertices, namely, donor vertices and tax vertices. A donor vertex v has a positive number $v.g$ associated with it, which is the amount of money that the traveller is gifted when he arrives at the vertex. The value of $v.g$ at a tax vertex v is negative and signifies the amount that the traveller must pay at this vertex. The source and target vertices have g values of 0. Find a path from the source to the target vertex such that the total expense along the path is minimum, where the total expense is the sum of the taxes levied by the tax vertices on the path minus the sum of gifted sums by the donor vertices on the path.

The input format is similar with the following change. Instead of the character R, D etc., in the beginning of each adjacency list, the number $v.g$ is given, which is positive for a donor vertex and negative for a toll vertex, and 0 for source or target vertices. (There may be other "neutral" vertices with g value of 0).

Notes: The graph given is directed, but may have positive cycles.