CS101 Project2 Q3

Jitender Jangra

April 2024

1 Problem: Money's Issue

In this we are asked to make a new problem on the provided dataset. Here goes the problem
Problem:
We are given a graph G in which each node represents a location in the city. Jeetu decides to go on a road trip with here friend from destination A to destination B(which are given as *source* and *target* in the code). Now there exist a shortest path between two nodes on the graph (if no such path exist, such cases are also handled). Now if they pass through a node on this shortest path Jeetu will have to spend money. In order to reduce the money he spends he say that he will pass thorough atmost k nodes on the shortest path and this path (which he takes) will have a path length less than or equal to twice the length of the shortest path.,so that he has to spend only at max at k nodes and also too much fuel is not to be utilised (by the condition on the path length). If there exists such path return YES and the path. If no such path exists the return NO.
Now help Jeetu to find such path, so that his money could be saved.

2 Solution:

Intuition:
This problem in essence want to say that we have two node A and B. Now on the dataset given to us, there will exist a shortest path between these two nodes (If no path exist between two nodes, such cases are also handled). Now we want another path that share at most K nodes with this shortest path (excluding starting node) along with the condition that this path's length should be less than or equal to the two time the length of the shortest path between these two nodes.
Now to solve this problem: the function is written in such a way that firstly it generates the shortest path between the *source* and *target*. If no path exist between these two it returns to main. Else it gets the shortest path length as well. Now it initialises a *queue* for modified BFS. The queue is initialised with a tuple $(source, [source])$ where first element is the current node, the second

elements is the current path. Now the *while* loop is basically a modified BFS where we start popping from the queue and see the neighbours of the current node. If the nodes are already included in current path then we overlook it ,else we add the node in the path..
And we update this info in the queue as well. Now the *if* part of this says that when the current node is the target node and the length of this path made is less than or equal to twice of the length of the shortest path and no of shared nodes ¡= k, ie this is the path we were looking for, thus it returns this path. (Now, as per my understanding there could be multiple such paths thus we can return any of such)

Note: The graph is made in the same way as made in the first question. Also I thought of this question by myself so there might be some edge cases which I am unable to think or consider.