Assignment 11

Samundra karki

May 2019

11.1

Your friend (who has not taken an "Algorithms and Data Structures" course) asks you for help on implementing an algorithm for finding the shortest path between two nodes u and v in a directed graph (possibly containing negative edge weights). The friend proposes the following algorithm:
1. Add a large constant to each edge weight such that all weights become positive.
2. Run Dijkstra's algorithm for the shortest path from u to v.
Prove or disprove the correctness of the above algorithm to find the shortest path (note that in order to disprove, you only need to give a counterexample).

Solution:

When a large constant is added to each edge of the weight then, the following things happen:
1) Addition of the of the large constant changes the value of weight of each edge and that change of the total weight of the all of the path.
2) When total weight of all of the path is changed the route for the shortest path is also changed which might lead to misleading answer i.e. the shortest path might not be the shortest one.

Counter Example:

Lets say a shortest path from A to with total wight is -2: **A** $\rightarrow$ **B** The path it takes is: **A** $\rightarrow$ **C(-3)** $\rightarrow$ **E(1)** $\rightarrow$ **B**–Here bracketed value represents the weight of each edge.

Other path was:
A $\rightarrow$ **F(2)** $\rightarrow$ **E(1)** $\rightarrow$ **B**
Now when we add a constant to eliminate the negative value we get a new graph and the path is change because the smallest path would be -3.
Original path from Dijkstra's algorithm:
A $\rightarrow$ **C(0)** $\rightarrow$ **E(4)** $\rightarrow$ **B**
Path due to addition of the a constant:
A $\rightarrow$ **F(-1)** $\rightarrow$ **E(4)** $\rightarrow$ **B**

This creates a contradiction with original shortest path, so we cannot conclude that this is the shortest path.