Introduction to Analysis of Algorithms | March 9, 2021
Boston University | CS 330
Professor Steve Homer, Dora Erdos | Spring 2021

Homework 5

Due ~~Wednesday March 10~~ Thursday March 11, 2021 at 11:59 pm ET on Gradescope

Problem 1. *20 points.*

Suppose you are given a directed graph G with edge weights $w(u, v)$ on each edge, and source s. Somebody has run Dijkstra's algorithm for you on this graph, and provides you with a length-n array $dist$, such that $dist[i]$ contains the weight of the shortest path from s to i. Further, they also give you the shortest paths tree T, which you may assume is in an adjacency list format.

As we know, running Dijkstra's algorithm on a graph takes time $\Theta(m \log(n))$ (wen using a binary heap based priority queue). Now, suppose that you have $dist$ and T available to you . The weight along one of the edges is decreased to $w'(u, v)$, i.e. $w'(u, v) \leq w(u, v)$. You may assume that $w'(u, v) \geq 0$.

1. Draw an example where the change in the edge weight doesn't change the distance of any of the nodes. Draw another example where it does. In your example you have to clearly indicate the weights, the updated weight and the distances before and after the change.
2. UPDATED! You want to find out, whether the decrease of the edge weight affects the lengths of the shortest paths at all. Design an algorithm that runs in $O(n)$ time and returns 'yes' if there is at least one node whose distance changes, 'no' if there is no change at all. (Note, that your algorithm doesn't have to compute updated distances or find all the nodes that change.) Don't forget to prove that your algorithm is correct.
3. In class we said that in the general case Dijkstra's algorithm fails if there are negative edge weights in the graph. (We even saw an example) However, as it turns out, it works if we allow the outgoing edges adjacent to the source s to have negative weights. Prove that in this case Dijkstra's is indeed correct.