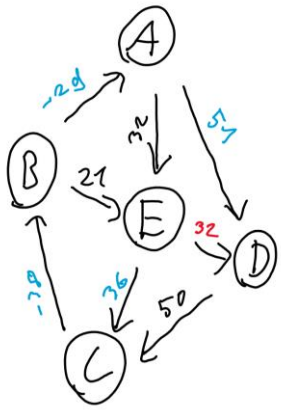
Problem 13.1

a)

This is in fact not true since adding at least the absolute value of the shortest path(negative) of the tree will change the graph to much overall.

Example:

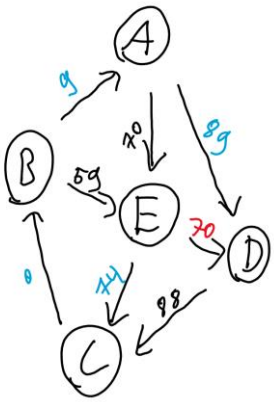


If we take this as our original graph, then shotrtest path from E to D is:

E->C->B->A->D with a total length of 20.

Another path would be E->D with a length of 32.

If we now make all edges positive by adding 38 to them, we get this tree:



Now the path E->C->B->A->D has a total length of 172 which is more than the path E->D which now has a length of 70. Therefore, this proves that just making all edges positive by addition gives us a different result than the unmodified tree.

b)

We can immediately disprove this if the tree has a negative weight cycle. Djikstra's Algorithm will get stuck in the negative weight loop since it will decrease the path from u to the current node indefinitely. Therefore the program will not terminate and will get stuck and never reach v.

Problem 13.3

a)

We can formalize the problem by thinking of it as a directed tree. The cells are the nodes, and the edges are the possible steps. This means that if we are on the starting cell in the top left corner, we have two possible options to go. It is either n steps down or n steps right. This means we have two outgoin edges from the current node to two the two nodes of the resulting cells. We give them the weight or length n of the current cell. Then we can just see if there is any path that connects the starting node to the ending node to see if there is a solution to the current board.