DSC 40B - Discussion 07

Problem 1.

What is the result of updating the edge (u,v) when the est[u], est[v] and weight(u,v) are given as follows?

Figure 1: Bellman Ford update subroutine

```
def update(u, v, weights, est, predecessor):
    if est[v] > est[u] + weights(u,v):
       est[v]=est[u]+weights(u,v)
       predecessor[v]=u
       return True
    else:
       return False
```

a) est[u] = 7, est[v] = 11, weight(u,v) = 3

b) est[u] = 15, est[v] = 12, weight(u,v) = -3

c) est[u] = 12, est[v] = 14, weight(u,v) = 3

Problem 2.

Run Bellman-Ford on the following graph using node s as the source. Below each node u, write the shortest path length from s to u. Mark the predecessor of u by highlighting it or making a bold arrow.

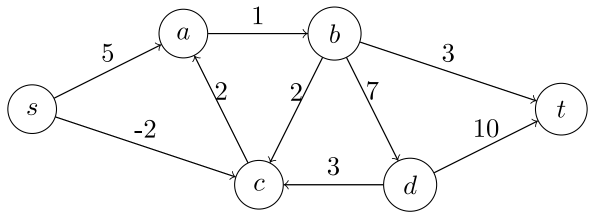
```
def bellman_ford(graph, weights, source):
    est={node:float('inf') for node in graph.nodes}
```

```
est[source]=0
predecessor={node: None for node in graph.nodes}
for i in range(len(graph.nodes)-1):
    for(u, v) in graph.edges:
        update(u, v, weights, est, predecessor)
return est, predecessor
```

Edge ordering convention. In each iteration (pass), perform `update(u, v)` exactly as listed below. Iteration pass uses this exact edge order:

$$(s,a),(s,c),(c,a),(a,b),(b,c),(b,d),(d,c),(b,t),(d,t).$$

Use this same edge order for every pass.



$$\begin{tikzcd}
 & a \arrow[r, "1"] & b \arrow[dl, "2"'] \arrow[dr, "7"] \arrow[rr, "3"] & & t \\
s \arrow[ur, "5"] \arrow[rr, "-2"'] & & c \arrow[ul, "2"] & d \arrow[l, "3"'] \arrow[ur, "10"] &
\end{tikzcd}$$

Problem 3.

Follow-up: Bellman-Ford example from Lecture 14 slide 46.

Consider the directed weighted graph from Lecture 14 (slide 46) with source node v_1. The weighted edges are:

$$(v_1, v_2) : 2,\ (v_2, v_3) : -1,\ (v_3, v_4) : 1,\ (v_4, v_5) : 3,\ (v_5, v_6) : 2,\ (v_1, v_7) : 7,\ (v_7, v_6) : 0,\ (v_7, v_5) : 3,\ (v_5, v_7) : -1.$$

Edge ordering convention. In each iteration (pass), perform `update(u, v)` exactly as listed below. Iteration pass uses this exact edge order:

$$(v_3, v_4),\ (v_1, v_2),\ (v_2, v_3),\ (v_7, v_6),\ (v_5, v_7),\ (v_7, v_5),\ (v_4, v_5),\ (v_5, v_6),\ (v_1, v_7).$$

Answer the following:

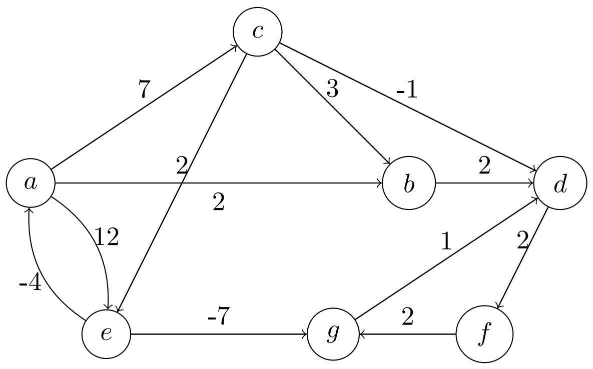
a) How many full passes are needed until distances no longer change?

b) What are the final `est` and `predecessor` dictionaries?

c) Briefly explain why this ordering is slower.

Problem 4.

a) Run Dijkstra's Algorithm on the following graph using node a as the source. Below each node u, write the shortest path length from a to u. Mark the predecessor of u by highlighting it or making a bold arrow.



b) Dijkstra's algorithm found the wrong path to some of the vertices. For just the vertices wherethe wrong path was computed, indicate both the path that was computed and the correct path.

c) What single edge could be removed from the graph such that Dijkstra's algorithm would happen to compute correct answers for all vertices in the remaining graph?