

---

## DSC 40B - Homework 08

Due: Wednesday, March 8

---

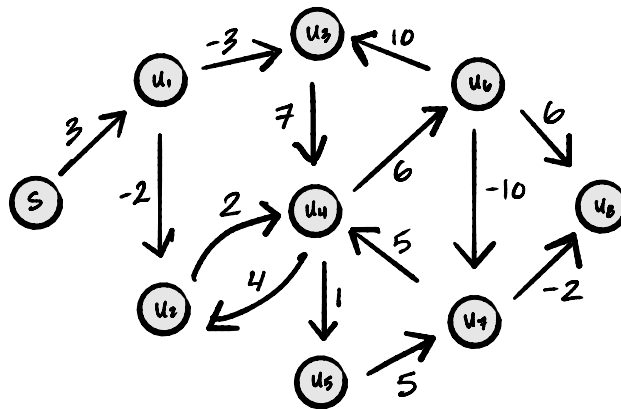
Write your solutions to the following problems by either typing them up or handwriting them on another piece of paper. Unless otherwise noted by the problem's instructions, show your work or provide some justification for your answer. Homeworks are due via Gradescope at 11:59 p.m.

**Note!** Because the midterm is on Thursday, we would like to release the solutions to this homework immediately after the due date on Wednesday at midnight. Since this is before the late due date, we **cannot accept slip days for this homework**.

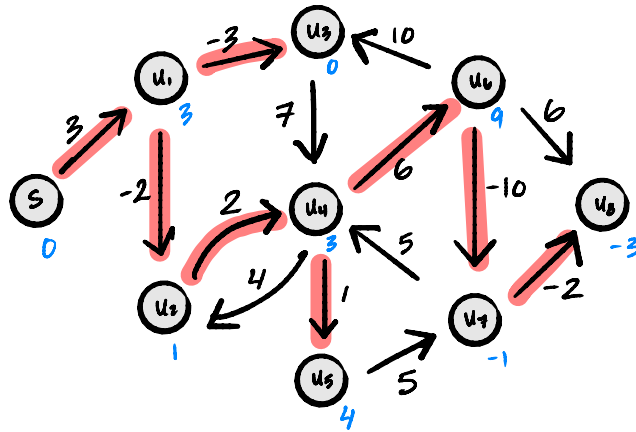
### Problem 1.

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. You can assume that `graph.edges` produces the graph's edges in the following order:

$(u_2, u_4), (u_7, u_8), (u_6, u_3), (s, u_1), (u_1, u_2), (u_4, u_5), (u_1, u_3),$   
 $(u_3, u_4), (u_6, u_7), (u_4, u_6), (u_5, u_7), (u_7, u_4), (u_4, u_2), (u_6, u_8)$



**Solution:**



### Problem 2.

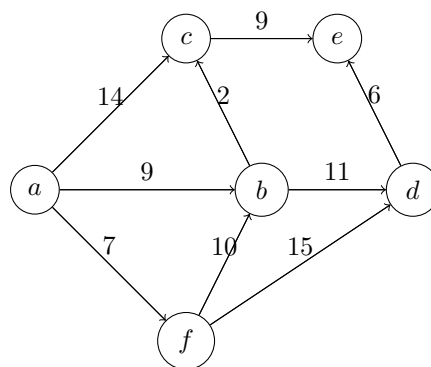
Recall that the Bellman-Ford algorithm (with early stopping) will terminate early if, after updating every edge, no predecessors have changed. Suppose it is known that the greatest shortest path distance in a graph  $G = (V, E)$  has  $\Theta(\sqrt{V})$  edges. What is the worst case time complexity of Bellman-Ford when run on this graph? State your answer using  $\Theta$  notation.

**Solution:** The loop invariant for Bellman-Ford says that after updating all edges  $\alpha$  times, all nodes whose shortest path distance is  $\leq \alpha$  has the correct shortest path distance. Since the longest shortest path is  $\Theta(\sqrt{V})$ , we'll find all shortest paths after  $\Theta(\sqrt{V})$  (outer) iterations. Every iteration involves  $\Theta(E)$  work because it updates every edge, so the total work of the loop is  $\Theta(E\sqrt{V})$ .

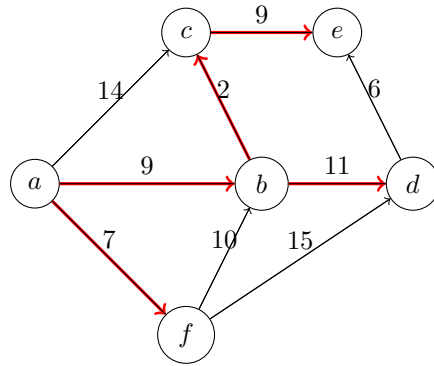
We also spend  $\Theta(V)$  time in the set up to initialize the estimated distance for each node in the graph. Therefore the total time is  $\Theta(V + \sqrt{V}E)$ .

### Problem 3.

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.



**Solution:**



est = { 'a': 0, 'b': 9, 'c': 11, 'd': 20, 'e': 20, 'f': 7 }

**Problem 4.**

True or False. Suppose  $G = (V, E, \omega)$  is a weighted graph for which all edges are positive, except for those edges of a node  $s$  which may or may not be negative. If Dijkstra's algorithm is run on  $G$  with  $s$  as the source, the correct shortest paths will be found. Assume that the graph does not have any negative loops. Justify your answer.

**Solution:** True. The reason that Dijkstra's algorithm can fail in the presence of negative edges is that it cannot rule out that a negative edge that it has not seen will cause a path which is currently longer to eventually become shorter. But if all of the negative edges are attached to the source, the rest of the edges are still positive; and so there is no way in which adding edges to a path can decrease its length.