Bellman-Ford algorithm

The Bellman-Ford algorithm

- Single-source shortest-paths problem
- Edge weights may be negative.
- Given a weighted, directed graph G = (V,E) with source s and weight function w
- The algorithm returns a Boolean value indicating whether or not there is a negative-weight cycle that is reachable from the source.
- If there is such a cycle, the algorithm indicates that no solution exists.
- If there is no such cycle, the algorithm produces the shortest paths and their weights.

Algorithm

```
BELLMAN-FORD(G, w, s)
1  INITIALIZE-SINGLE-SOURCE(G, s)
2  for i = 1 to |G.V| − 1
3       for each edge (u, v) ∈ G.E
4            RELAX(u, v, w)
5  for each edge (u, v) ∈ G.E
6       if v.d > u.d + w(u, v)
7            return FALSE
8  return TRUE
```

initializing the d and π values of all vertices

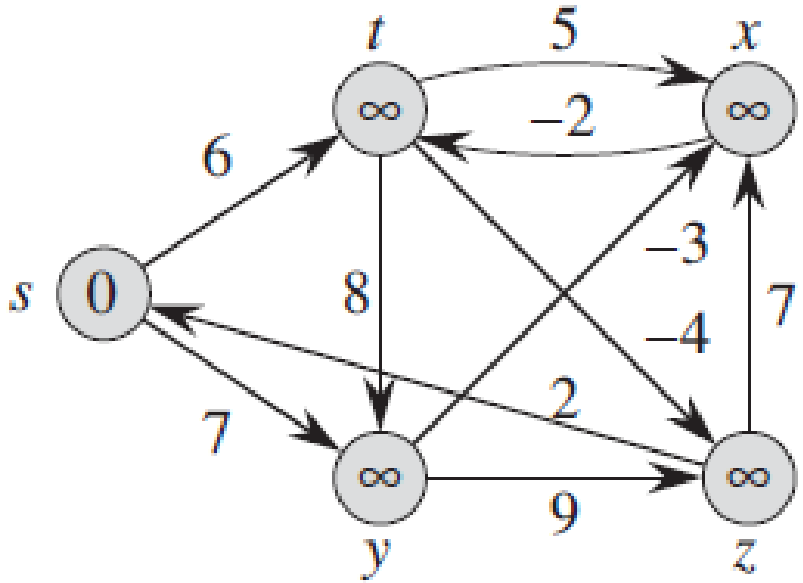
$|V| - 1$ passes over the edges of the graph

check for a negative-weight cycle

```
RELAX(u, v, w)
1  if v.d > u.d + w(u, v)
2       v.d = u.d + w(u, v)
3       v.π = u
```

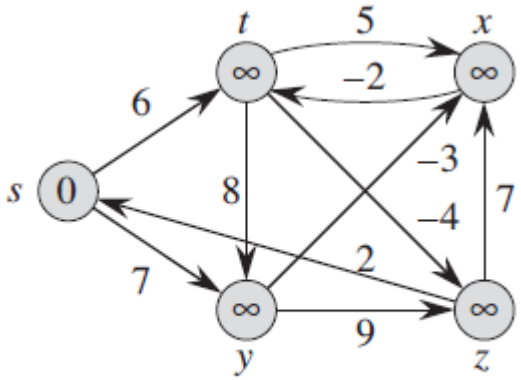
The Bellman-Ford algorithm runs in time $O(VE)$

Example

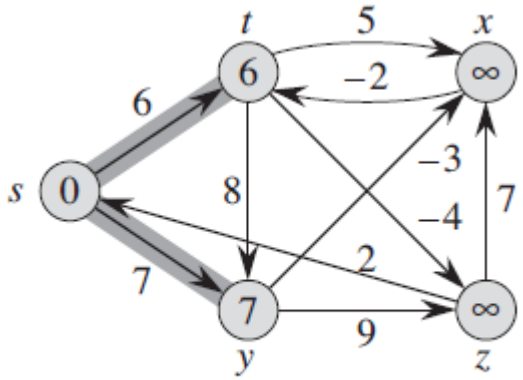


Each pass relaxes the edges in the order

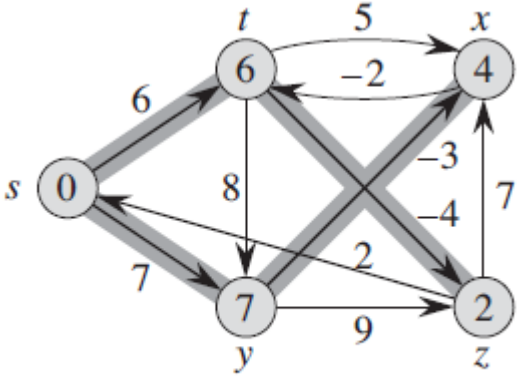
$$(t,x),(t,y),(t,z),(x,t),(y,x),(y,z),(z,x),(z,s),(s,t),(s,y).$$



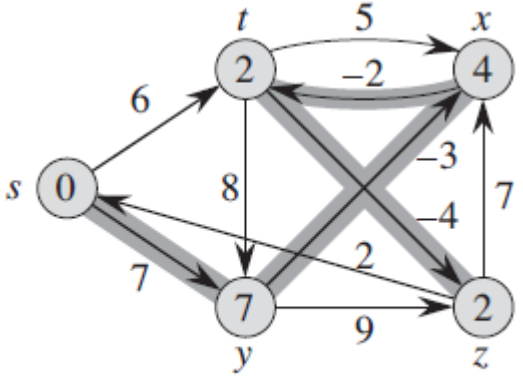
(a)



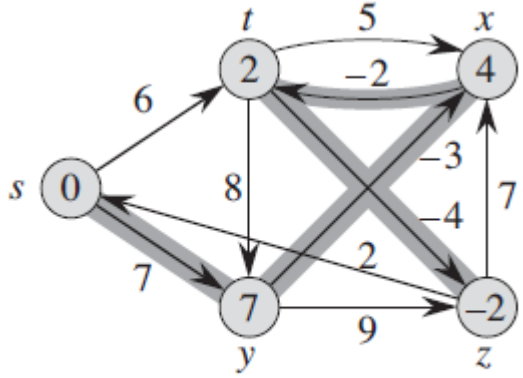
(b)



(c)



(d)



(e)

returns TRUE if graph G contains no negative-weight cycles

Homework

1. Find the shortest path tree from every node to node 1 for the graph of Fig.1 using the Bellman-ford

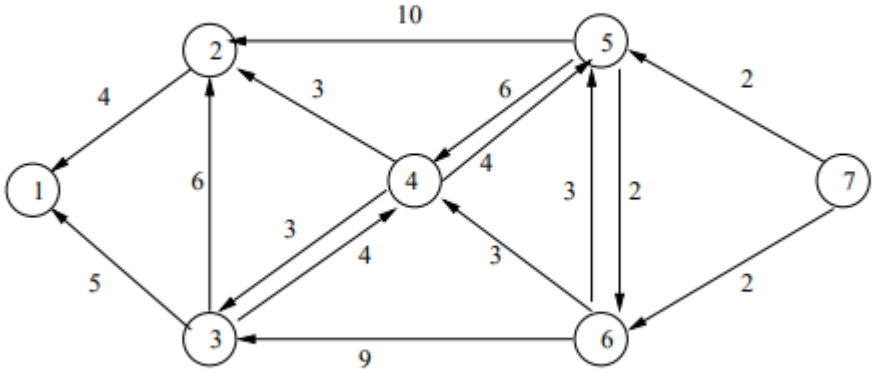


Figure 1: Graph for Problem 1

Homework

- Check negative cycle in the given graph