CSCI 532 – Algorithm Design
Assignment 10

Name: Badarika Namineni **CWID:** 50233279

Question 1. Run the Bellman-Ford algorithm on the directed graph of Figure 24.4, using vertex z as the source. In each pass, relax edges in the same order as in the figure, and show the d and π values after each pass. Now, change the weight of edge (z, x) to 4 and run the algorithm again, using s as the source.

Solution:

	S	T	X	y	z
d	2	4	6	9	0
π	Z	X	Y	z	NIL

	S	T	X	y	z
d	0	0	2	7	-2
π	NIL	X	Z	S	t

Question 2. Modify the Bellman-Ford algorithm so that it sets v.d to _∞ for all vertices v for which there is a negative-weight cycle on some path from the source to v.

Solution:

```
Bellman-Ford-New(G, w, s)
        Initialize-Single-Source(G, s)
        for i <- 1 to |V[G]| - 1
                do for each edge (u,v) ∈ E[G]
                        do Relax[u, v, w]
        for each edge (u, v) ∈ E[G]
                do if d[v] > d[u] + w(u, v)
                        d[v] = -∞
        for each v such that d[v] = -∞
                do Follow-And-Mark-Pred(v)

Follow-And-Mark-Pred(v)
        if π[v] != nil and d[π[v]] != -∞
                do d[π[v]] = -∞
                Follow-And-Mark-Pred(π[v])
        else
                return
```

Question 3. Suppose we change line 3 of DAG-SHORTEST-PATHS to read 3 for the first |V| - 1 vertices, taken in topologically sorted order

Show that the procedure would remain correct.

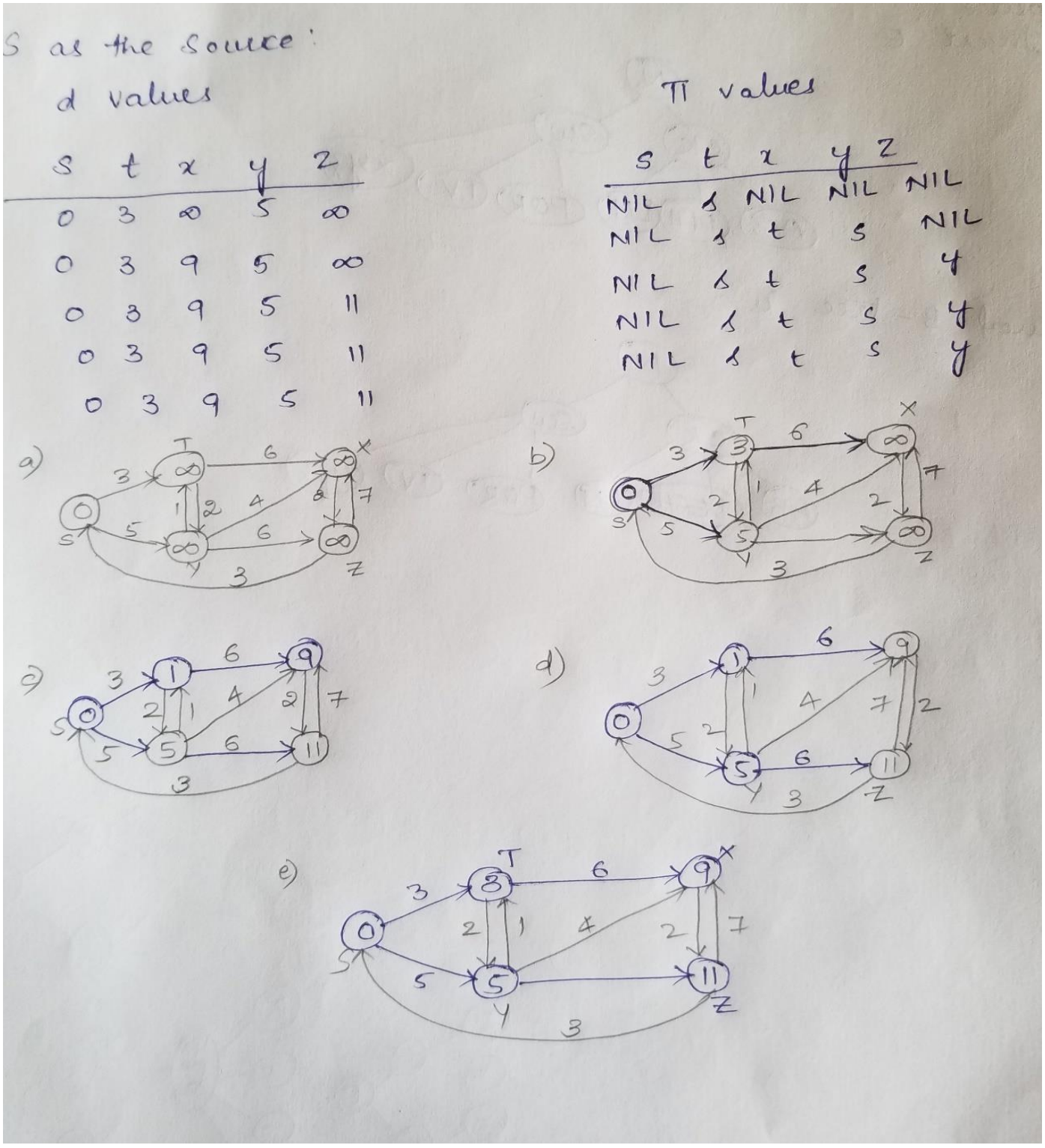
Solution:

When we reach vertex v, the last vertex in the topological sort, it must have out degree 0. Otherwise there would be an edge pointing from a later vertex to an earlier vertex in the ordering,

which is a contradiction. Thus, the body of the for-loop of line 4 is never entered for this final vertex.

Question 4. Run Dijkstra's algorithm on the directed graph of Figure 24.2, first using vertex s as the source and then using vertex z as the source. In the style of Figure 24.6, show the d and π values and the vertices in set S after each iteration of the while loop.

Solution:



z as the source

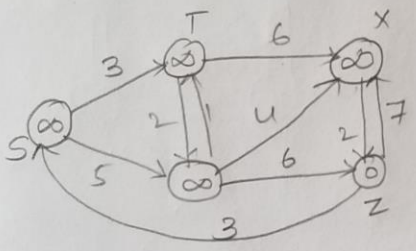
d values

s	t	x	y	z
3	∞	7	∞	0
3	6	7	8	0
3	6	7	8	0
3	6	7	8	0
3	6	7	8	0

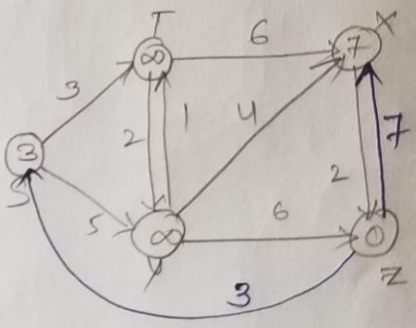
Π values

s	t	x	y	z
z	NIL	z	NIL	NIL
z	s	z	s	NIL
z	s	z	s	NIL
z	s	z	s	NIL
z	s	z	s	NIL

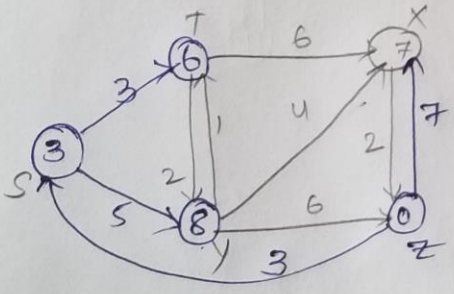
a)



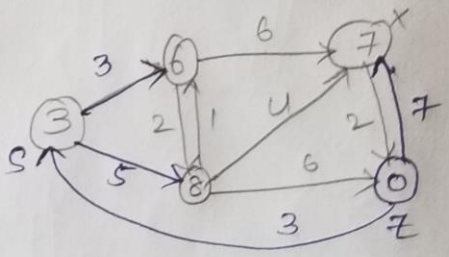
b)



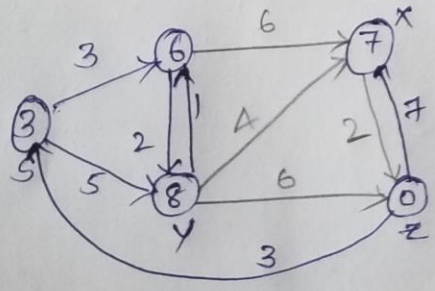
c)



d)



e)



Question 5. Give a simple example of a directed graph with negative-weight edges for which Dijkstra's algorithm produces incorrect answers. Why doesn't the proof of Theorem 24.6 go through when negative-weight edges are allowed?

Solution:

Consider any graph with a negative cycle. RELAX is called a finite number of times but the distance to any vertex on the cycle is $-\infty$, so Dijkstra's algorithm cannot possibly be correct here. The proof of theorem 24.6 doesn't go through because we can no longer guarantee that $\delta(s,y) \leq \delta(s,u)$.

Question 6. Professor Gang Reath thinks that he has worked out a simpler proof of correctness for Dijkstra's algorithm. He claims that Dijkstra's algorithm relaxes the edges of every shortest path in the graph in the order in which they appear on the path, and therefore the path-relaxation property applies to every vertex reachable from the source. Show that the professor is mistaken by constructing a directed graph for which Dijkstra's algorithm could relax the edges of a shortest path out of order.

Solution:

Let the graph have vertices s, x, y, z and edges (s, x), (x,y), (y,z), (s,y), and let every edge have weight 0. Dijkstra's algorithm could relax edges in the order (s,y), (s,x), (y,z), (x,y). The graph has two shortest paths from s to z: (s, x, y, z) and (s, y, z) both with weight 0. The edges on the shortest path (s, x, y, z) are relaxed out of order, because (x,y) is relaxed after (y,z).