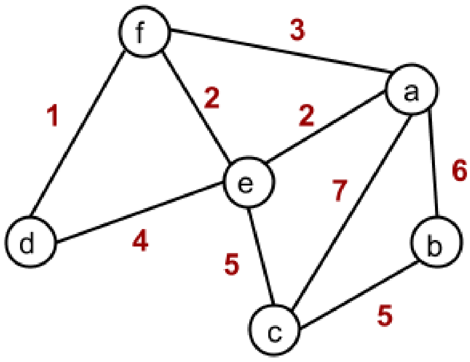
HW12 Solutions

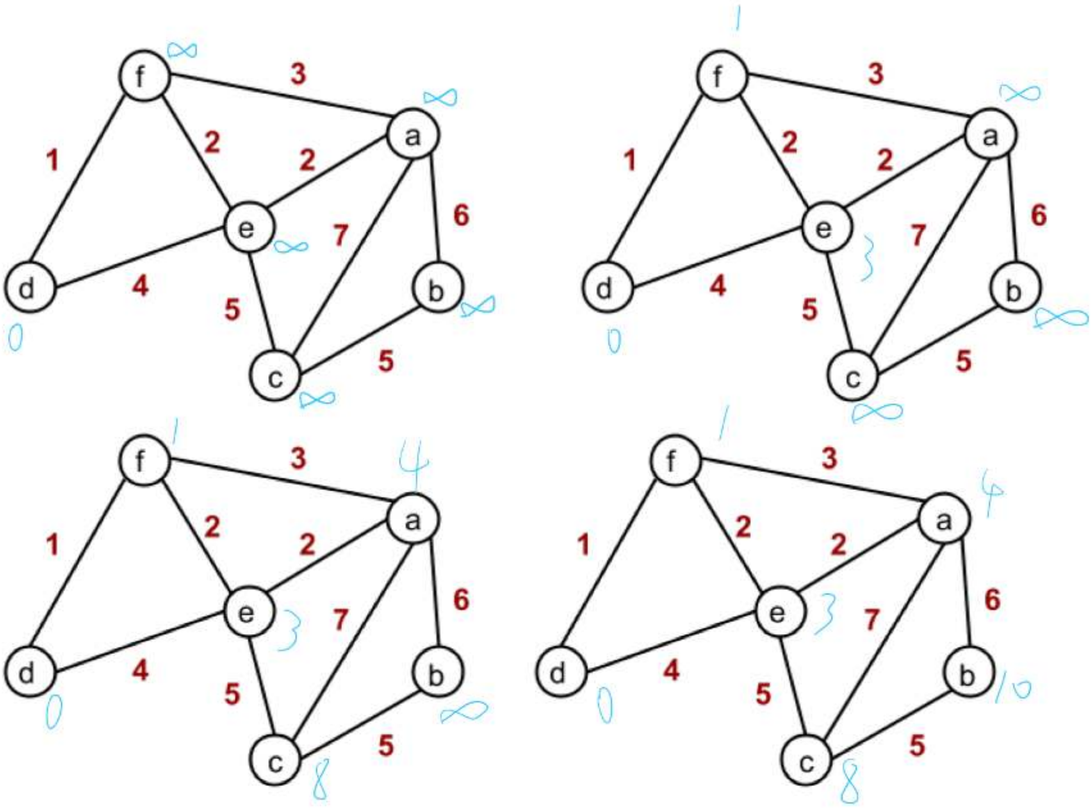
EL9343 Homework 12

Due: Dec. 14th 8:00 a.m.

1. Run the Bellman-Ford algorithm on the following graph, with vertex d as the source. In each pass, relax the edges in the order of $\{(a,b),(a,c),(a,e),(a,f),(b,c),(c,e),(d,e),(d,f),(e,f)\}$. Write down the d array after each pass.

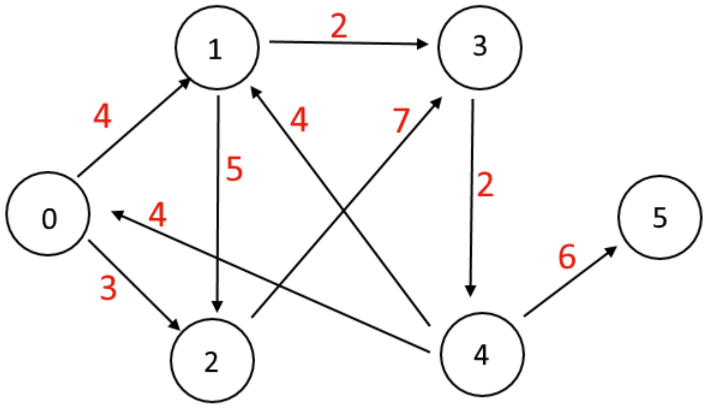


Solution:



No.	a	b	c	d	e	f
1	∞	∞	∞	0	∞	∞
2	∞	∞	∞	0	3	1
3	4	∞	8	0	3	1
4	4	10	8	0	3	1

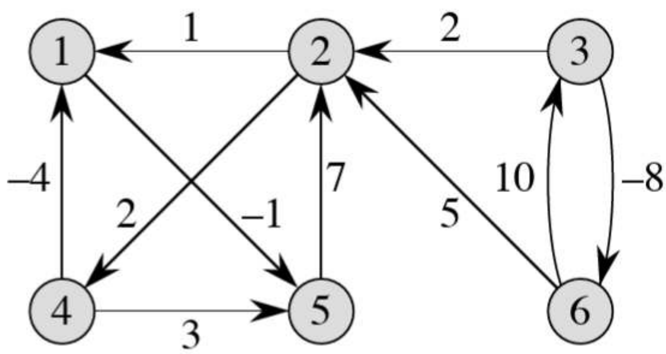
2. Run the Dijkstra's algorithm on the following graph, with node 0 as the source node. Write down the d array before each EXTRACT-MIN, also the final array.



Solution:

No.	0	1	2	3	4	5
1	0	∞	∞	∞	∞	∞
2	0	4	3	∞	∞	∞
3	0	4	3	10	∞	∞
4	0	4	3	6	∞	∞
5	0	4	3	6	8	∞
6	0	4	3	6	8	14

3. Run the Floyd-Warshall algorithm on the following graph. Show the matrix $D^{(k)}$ that results for each iteration of the outer loop.



Solution:

k	D^k
0	$\begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & \infty & \infty \\ \infty & 2 & 0 & \infty & \infty & -8 \\ -4 & \infty & \infty & 0 & 3 & \infty \\ \infty & 7 & \infty & \infty & 0 & \infty \\ \infty & 5 & 10 & \infty & \infty & 0 \end{pmatrix}$
1	$\begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & 0 & \infty \\ \infty & 2 & 0 & \infty & \infty & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ \infty & 7 & \infty & \infty & 0 & \infty \\ \infty & 5 & 10 & \infty & \infty & 0 \end{pmatrix}$
2	$\begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & 0 & \infty \\ 3 & 2 & 0 & 4 & 2 & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ 8 & 7 & \infty & 9 & 0 & \infty \\ 6 & 5 & 10 & 7 & 5 & 0 \end{pmatrix}$
3	$\begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & 0 & \infty \\ 3 & 2 & 0 & 4 & 2 & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ 8 & 7 & \infty & 9 & 0 & \infty \\ 6 & 5 & 10 & 7 & 5 & 0 \end{pmatrix}$
4	$\begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ 0 & 2 & 0 & 4 & -1 & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{pmatrix}$
5	$\begin{pmatrix} 0 & 6 & \infty & 8 & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ 0 & 2 & 0 & 4 & -1 & -8 \\ -4 & 2 & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{pmatrix}$
6	$\begin{pmatrix} 0 & 6 & \infty & 8 & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ -5 & -3 & 0 & -1 & -6 & -8 \\ -4 & 2 & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{pmatrix}$

4. We are given a directed graph $G = (V, E)$ on which each edge $(u, v) \in E$ has an associated value $r(u, v)$, which is a real number in the range $0 \leq r(u, v) \leq 1$ that represents the reliability of a communication channel from vertex u to vertex v. We interpret $r(u, v)$ as the probability that the channel from u to v will not fail, and we assume that these probabilities are independent. Give an efficient algorithm to find the most reliable path between two given vertices, u and v. Assume that Dijkstra's algorithm runs in $O(|E| \log |V|)$, and your algorithm should also run in $O(|E| \log |V|)$. Briefly describe why your algorithm is correct.

   Similar to Dijkstra's algorithm, the time complexity is $O(|E| log |V|)$.
   Another way is to define a new weight function. Let $c(u, v) = -\log r(u, v)$. Then we can show that, finding the most reliable path is equivalent to finding the shortest path in G if the length of each edge

Algorithm 1 Find The Most Reliable Path

Input: graph: G = (V,E)
Input: source and target vertex: s, t
Input: probability matrix: $r(u, v) \in R$
Output: Most Reliable Path

```
for v ∈ V do
  p[v] = 0
end for
p[s] = 1
S = {}
Q = MAX − HEAPIFY(V)
while Q ≠ ∅ do
  u = EXTRACT − MAX(Q)
  for v ∈ Adj[u] do
    if p[v] < p[v] × r(u, v) then
      p[v] = p[v] × r(u, v)
    end if
  end for
end while
```

e is $c(e)$. For the most reliable path, we are to find a path p such that the total reliability is maximized, i.e. $\max_p[\prod_{\{e|e \text{ is in } p\}} r(e)]$. For the shortest path case, we want $\min_p[\sum_{\{e|e \text{ is in } p\}}(-\log r(e))] = \min_p[\sum_{\{e|e \text{ is in } p\}} c(e)]$. It is obvious that the two problem share a same solution.