THE BELLMAN – FORD ALGORITHM

CSE [AIML], 6TH SEMESTER

DESIGN & ANALYSIS OF ALGORITHMS

INTRODUCTION

The algorithm solves the single source shortest paths problem in which edge weights may be negative.

Given a weighted graph G = (v, e) with source s and weight function w: e -> R. 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.

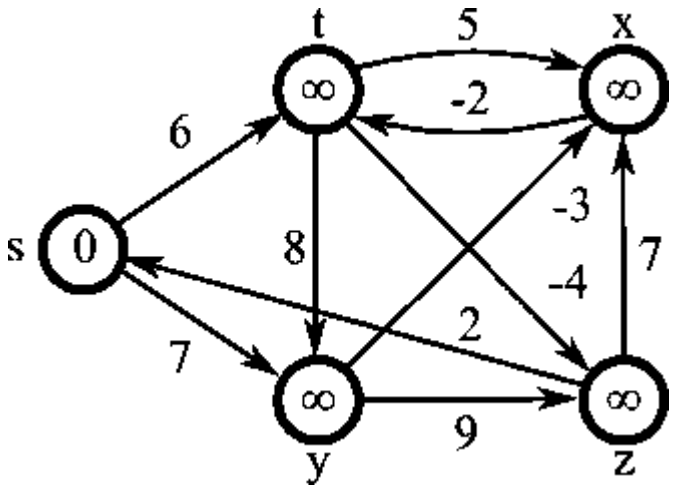
ALGORITHM

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) ϵ G.E
4. RELAX(u, v, w)
5. For each edge(u,v) ϵ G.E
6. If v.d > u.d + w(u,v)
7. Return false
8. Return true

Example

- Given a weighted graph with V vertices and E edges, and a source vertex s, find the shortest path from the source vertex to all vertices in the given graph.



Solution

- Number of vertices = 5
- Weight Matrix

	s	t	x	y	z
s	0	6	Inf	7	inf
t	inf	0	5	8	-4
x	inf	-2	0	inf	inf
y	inf	inf	-3	0	9
z	2	inf	7	inf	0

EDGE	WEIGHT
s-t	6
s-y	7
t-x	5
t-y	8
t-z	-4
x-t	-2
y-x	-3
y-z	9
z-s	2
z-x	7

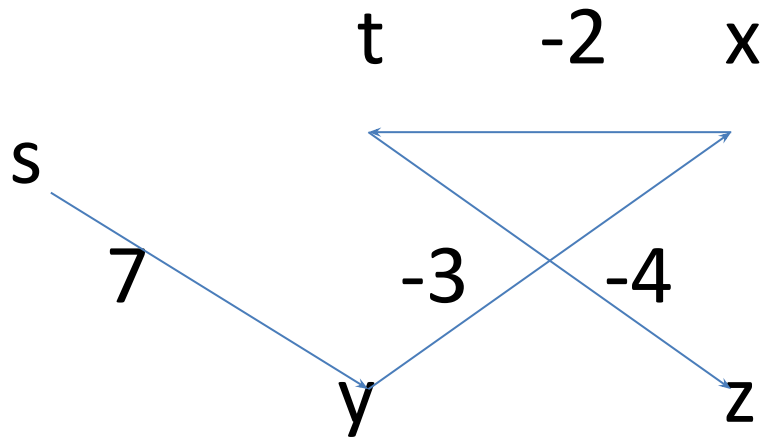
Solution continues…

- Distance Matrix Predecessor Matrix

d	s	t	x	y	z
	0	inf	inf	inf	inf
1	**0**	**6**	inf	7	inf
2	**0**	**6**	11	7	**2**
3	**0**	**6**	9	**7**	**2**
4	**0**	**6**	**4**	**7**	**2**

П	s	t	x	y	z
-	-	-	-	-	-
1	-	s	-	s	-
2	-	s	t	s	t
3	-	s	z	s	t
4	-	s	y	s	t

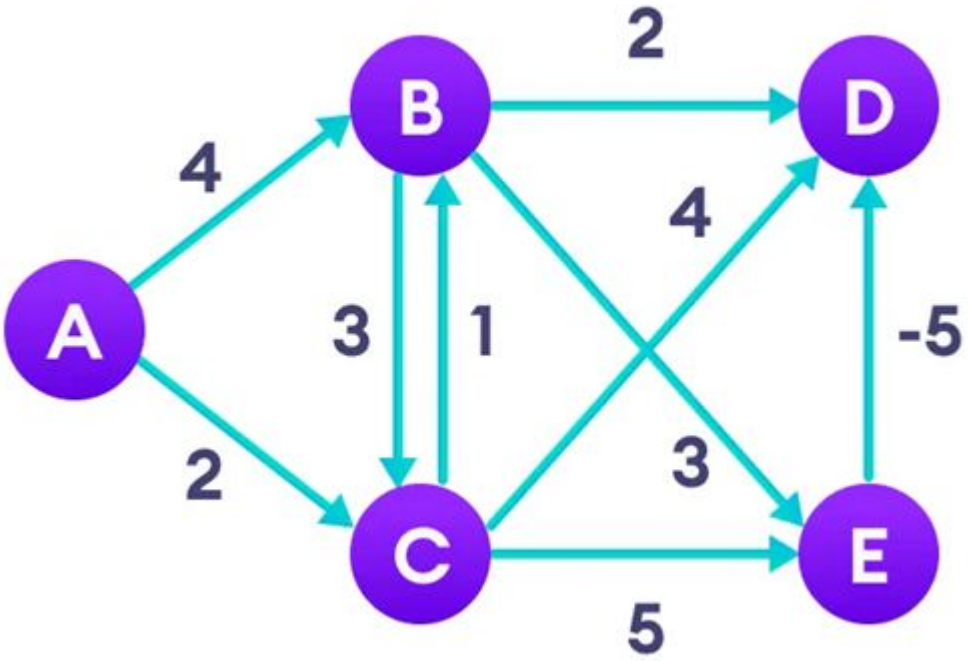
SHORTEST PATH



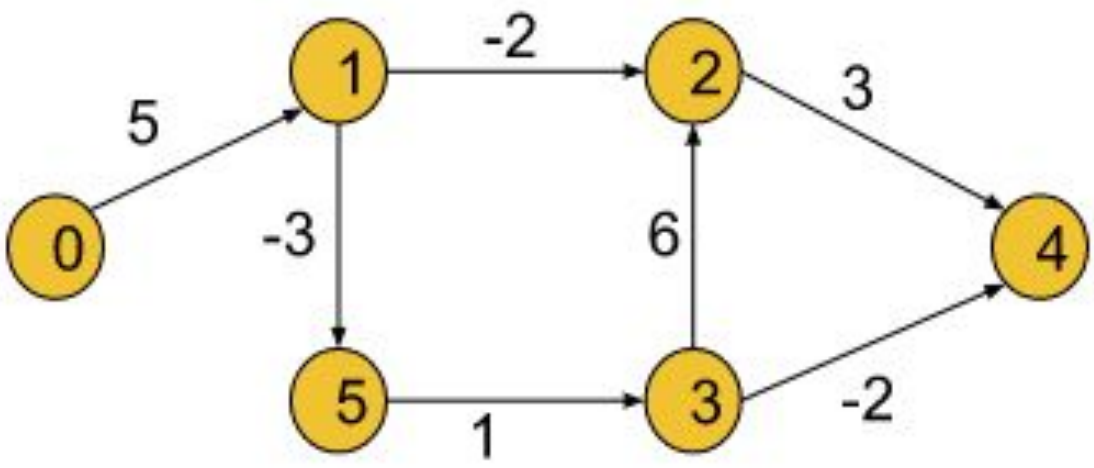
Complexity

The algorithm runs in time O(VE), since the initialization in line 1 takes O(V) time, each of the |V|-1 passes over the edges in lines 2-4 take O(E) time and the for loop of lines 5-7 takes O(E) time

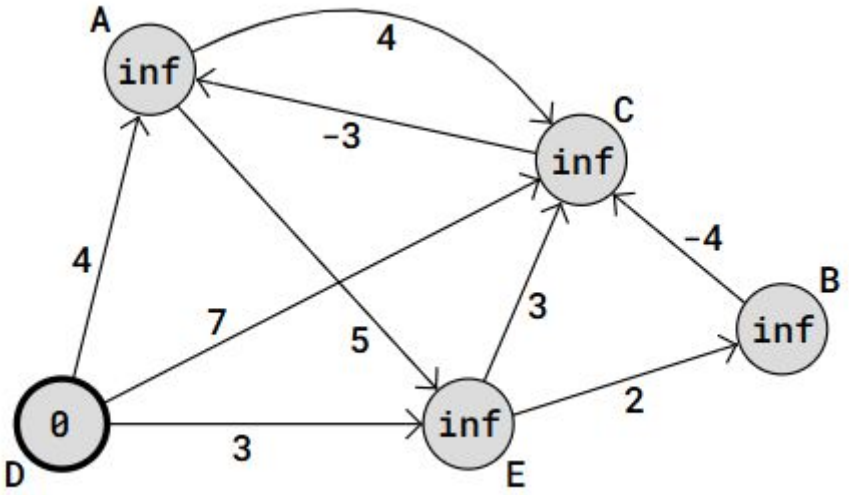
Exercise 1



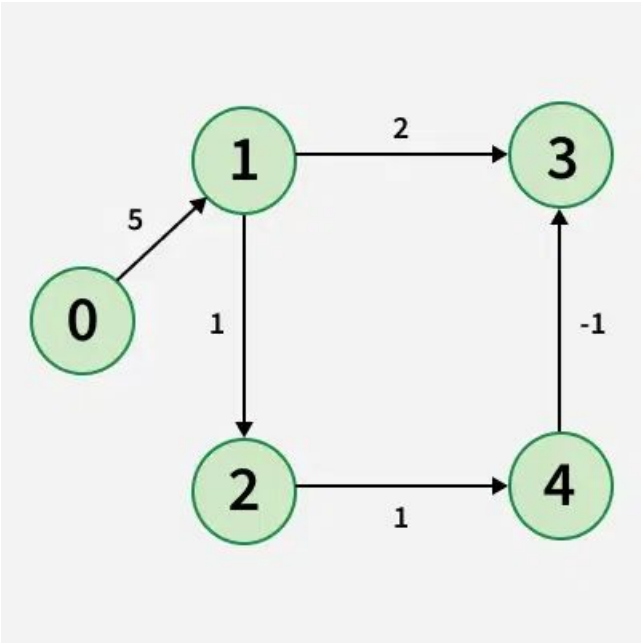
Exercise 2



Exercise 3



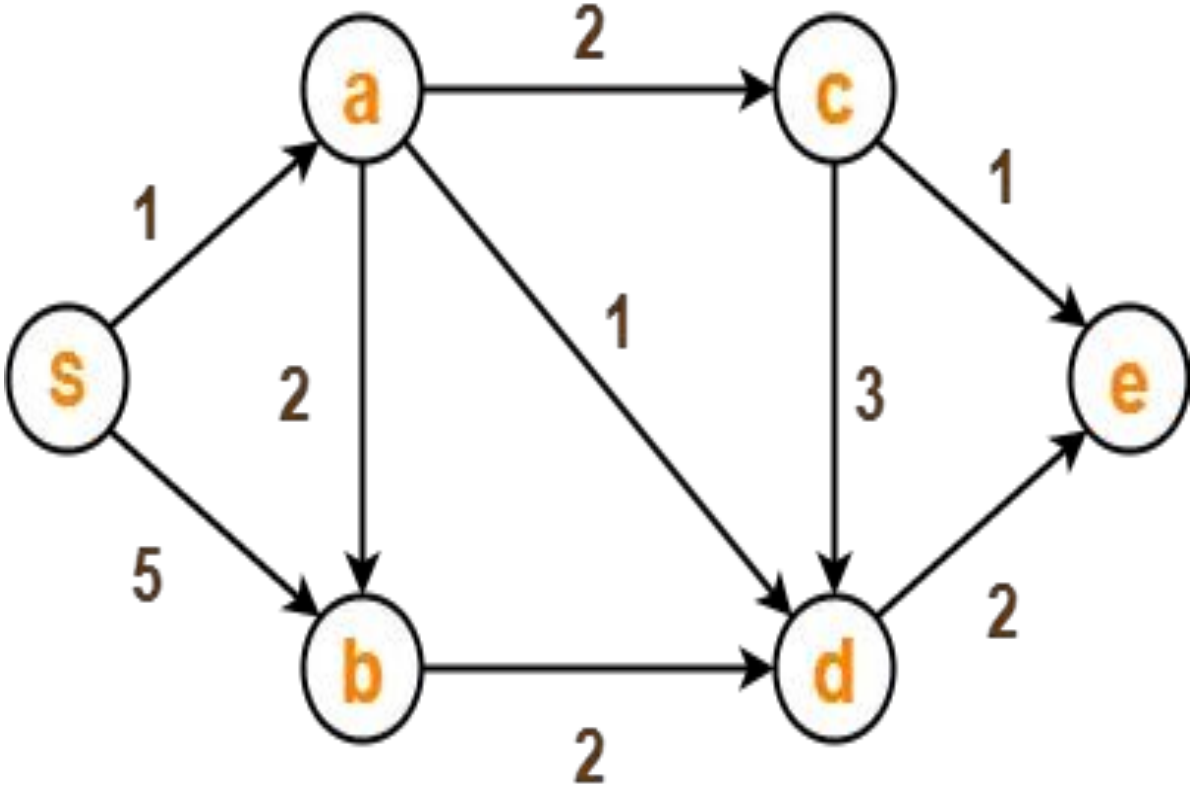
Exercise 4



DIJKSTRA’S ALGORITHM

EXAMPLE

Example1



EDGE	WEIGHT
s-a	1
s-b	5
a-b	2
a-c	2
a-d	1
b-d	2
c-d	3
c-e	1
d-e	2

Solution

DISTANCE MATRIX						
d	s	a	b	c	d	e
0	0	α	α	α	α	α
1	**0**	**1**	5	α	α	α
2	**0**	**1**	3	3	**2**	α
3	**0**	**1**	**3**	3	**2**	4
4	**0**	**1**	**3**	**3**	**2**	4
5	**0**	**1**	**3**	**3**	**2**	**4**
6	**0**	**1**	**3**	**3**	**2**	**4**

PREDECESSOR MATRIX						
π	s	a	b	c	d	e
0	-	-	-	-	-	-
1	-	s	s	-	-	-
2	-	s	a	a	a	-
3	-	s	a	a	a	d
4	-	s	a	a	a	d
5	-	s	a	a	a	d
6	-	s	a	a	a	d

Shortest path

s,a,d,b,c,e