AIUB
SOLUTION

SUBJECT - ALGORITHM

FINAL TERM HAND NOTE

SUMMER 2019

SOLVED BY- MUSTAFIZUR RAHMAN

X = ABC BDAB → 7

Y = BDC ABA → 6

BCBA

LCS

		0	1	2	3	4	5	6
			B	D	C	A	B	A
	0	0	0	0	0	0	0	0
A	1	0	0↑	0↑	0↑	1↖	1←	1↖
B	2	0	1↖	1←	1←	1↑	2↖	2←
C	3	0	1↑	1↑	2↖	2←	2↑	2↑
B	4	0	1↖	1↑	2↑	2↑	3↖	3←
D	5	0	1↑	2↖	2↑	2↑	3↑	3↑
A	6	0	1↑	2↑	2↑	3↖	3↑	4↖
B	7	0	1↖	2↑	2↑	3↑	4↖	4↑

- মিললে উপরের কোনাকোনি ঘরের সাথে 1 যোগ ↖
- না মিললে check
  - ⇒ উপরে বড় হলে ↑
  - ⇒ সমান হলে ↑
  - ⇒ ছোট হলে ←

B C B A

X = a b c d c d a b   bca

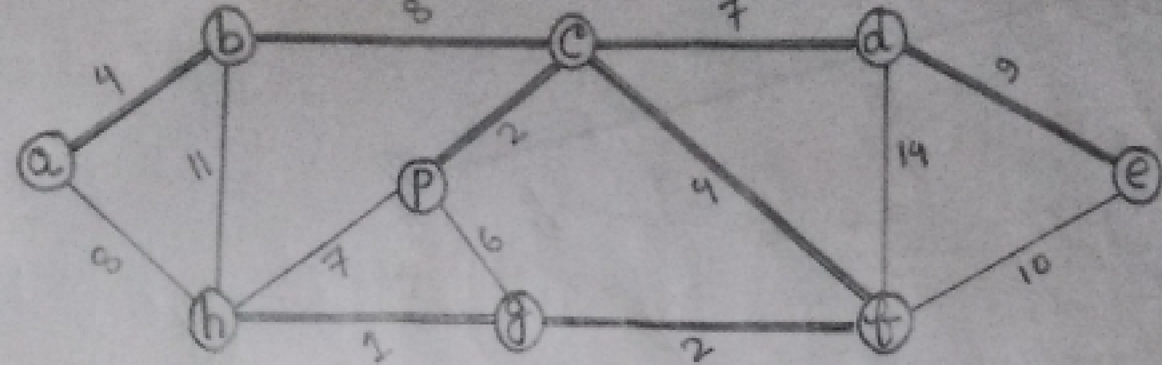
Y = d c b c a a

	0	a	b	c	d	c	d	a	b
0	0	0	0	0	0	0	0	0	0
d	0	0↑	0↑	0↑	1↖	1←	1↖	1←	1←
c	0	0↑	0↑	1↖	1↑	2↖	2←	2←	2←
b	0	0↑	1↖	1↑	1↑	2↑	2↑	2↑	3↖
c	0	0↑	1	2↖	2←	2↖	2↑	2↑	3↑
a	0	1↖	1↑	2↑	2↑	2↑	2↑	3↖	3↑
a	0	1↑	1↑	2↑	2↑	2↑	2↑	3↑	3↑

b c a

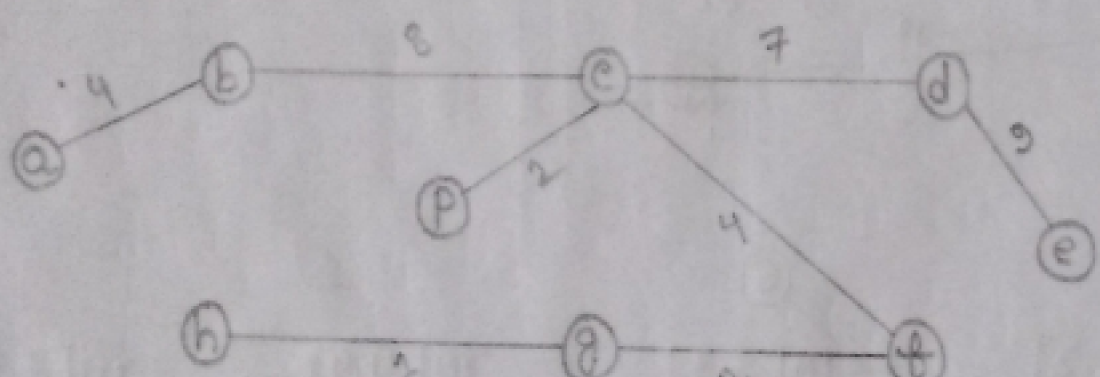
Minimum Spanning Tree

Kruskal Algorithm



I) Find minimum spanning tree.

II) Find minimum cost.

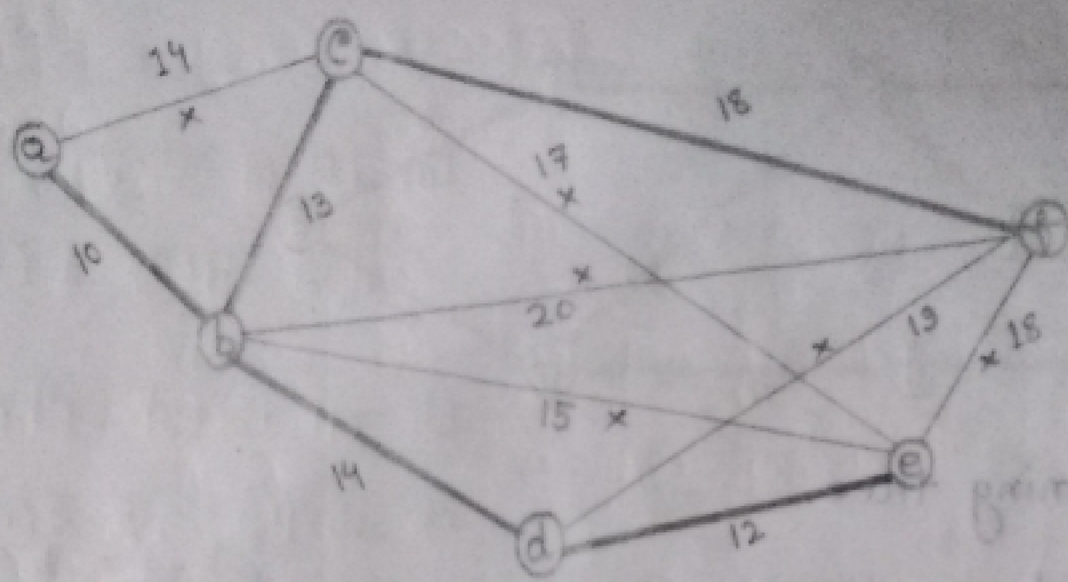


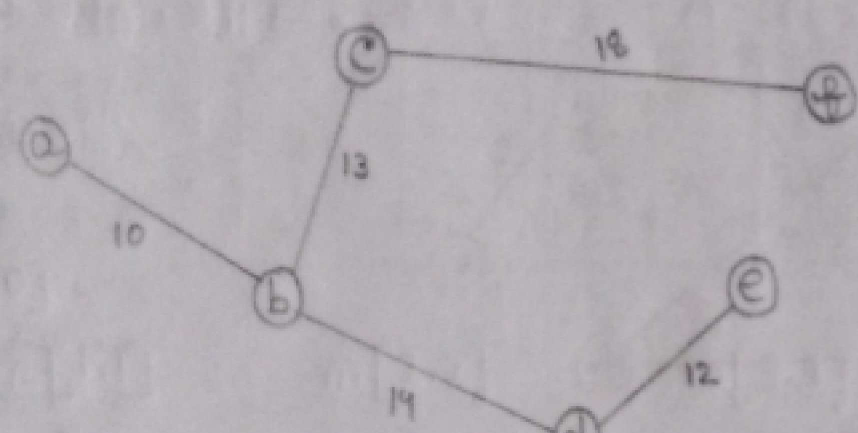
h-g = 1 ✓
c-p = 2 ✓
g-f = 2 ✓
c-f = 4 ✓
a-b = 4 ✓
p-g = 6 ✗
c-d = 7 ✓, h-p = 7 ✗
b-c = 8 ✓
a-h = 8 ✗
d-e = 9 ✓
f-e = 10 ✗
b-h = 11 ✗
d-f = 14 ✗

minimum cost = 4+8+7+9+2+2+4+1

= 37

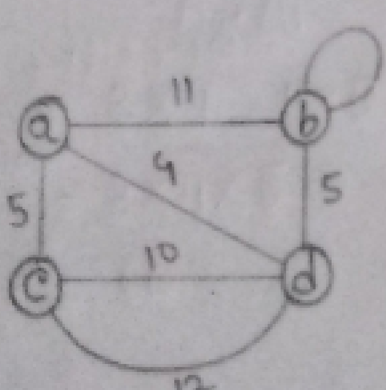
Prim's Algorithm





weight $= 10+14+12+13+18$

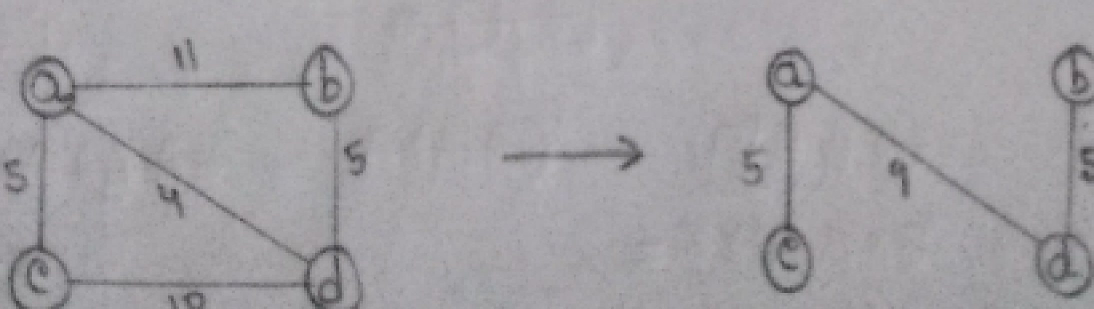
$= 67$



In case of these type of graph -

i) remove loop

ii) remove parallel line(greater one)



$= 14$

Matrix Chain Multiplication

A_0	A_1	A_2	A_3	A_4
3X4	4X5	5X6	6X2	2X5
P_1 P_2	P_1 P_2	P_1 P_2	P_1 P_2	P_1 P_2

$A_0 = 3X4$ $\qquad$ $A_3 = 6X2$

$A_1 = 4X5$ $\qquad$ $A_4 = 2X5$

$A_2 = 5X6$

	A_0 0	A_1 1	A_2 2	A_3 3	A_4 4
A_0 0	0	60	150	124	154
A_1 1		0	120	100	140
A_2 2			0	60	110
A_3 3				0	60
A_4 4					0

Step 1 : m[0,0] A_0 m[1,1] A_1 m[2,2] A_2 m[3,3] A_3 m[4,4] A_4 ~~m[5,5]~~ ~~A_5~~

Step 2 :

m[0,1]	m[1,2]	m[2,3]	m[3,4]	~~m[4,5]~~
$A_0 \cdot A_1$	$A_1 \cdot A_2$	$A_2 \cdot A_3$	$A_3 \cdot A_4$	~~$A_4 \cdot A_5$~~
3X4 4X5	4X5 5X6	5X6 6X2	6X2 2X5	~~m[5,6]~~
= 3X4X5	= 4X5X6	= 5X6X2	= 6X2X5	
= 60	= 120	= 60	= 60	

Step 3 : m[0,2]

$A_0(\underbrace{A_1 \cdot A_2}_{X})$

= 4X5 5X6 + $A_0 \cdot X$

= 4X6 + 3X4 4X6

= 4X5X6 + 3X4X6

= 120 + 72 = 192

$(A_0 \cdot A_1) A_2$

= 3X4 4X5 + 3X5 5X6

= 3X4X5 + 3X5X6

= 60 + 90

= 150

$m[1,3]$

$A_1 (A_2 \cdot A_3)$

$= 5\times6 \;\; 6\times2 + 4\times5 \;\; 5\times2$

$= 5\times6\times2 + 4\times5\times2$

$= 60+40$

$= 100$

$(A_1 A_2) A_3$

$= 4\times5 \;\; 5\times6 + 4\times6 \;\; 6\times2$

$= 4\times5\times6 + 4\times6\times2$

$= 120+48$

$= 168$

$m[2,4]$

$A_2 (A_3 \cdot A_4)$

$= 6\times2 \;\; 2\times5 + 5\times6 \;\; 6\times5$

$= 6\times2\times5 + 5\times6\times5$

$= 60+150$

$= 210$

$(A_2 A_3) A_4$

$= 5\times6 \;\; 6\times2 + 5\times2 \;\; 2\times5$

$= 5\times6\times2 + 5\times2\times5$

$= 60+50$

$= 110$

Step 4 :-

$$\boxed{M[i,j] = M[i,K] + M[K+1,j] + P_i * P_{K+1} * P_{j+1}}$$

$m[\underset{i}{0},\underset{j}{3}]$

$K=0$

$A_0 \cdot (A_1 A_2 A_3)$

$= m[0,0] + m[1,3] + 3\times4\times2$

$= 0+100+24$

$= 124$

P_i = first dimension of first partition

P_{K+1} = first dimension of second partition

P_{J+1} = last dimension of second partition

$K=1$

$(A_0 A_1) \cdot (A_2 A_3)$

$= m[0,1] + m[2,3] + 3 \times 5 \times 2$

$= 60+60+30$

$= 150$

$K=2$

$(A_0 A_1 A_2) A_3$

$= m[0,2] + m[3,3] + 3 \times 6 \times 2$

$= 150+0+36$

$= 186$

$m[1,4]$

$K=1$

$A_1 (A_2 \cdot A_3 \cdot A_4)$

$= m[1,1] + m[2,4] + 4 \times 5 \times 5$

$= 0+110+100$

$= 210$

$K=2$

$(A_1 A_2)(A_3 A_4)$

$= m[1,2] + m[3,4] + 4 \times 6 \times 5$

$= 120+60+120$

$= 300$

$K=3$

$(A_1 A_2 A_3) A_4$

$= m[1,3] + m[4,4] + 4 \times 2 \times 5$

$= 100+0+40$

$= 140$

Step-5

$m[0,4]$

$K=0$

$A_0 (A_1 A_2 A_3 A_4)$

$= m[0,0] + m[1,4] + 3 \times 4 \times 5$

$= 0 + 140 + 60$

$= 200$

$K=1$

$(A_0 A_1)(A_2 A_3 A_4)$

$= m[0,1] + m[2,4] + 3 \times 5 \times 5$

$= 60 + 110 + 75$

$= 245$

$K=2$

$(A_0 A_1 A_2) A_3 A_4$

$= m[0,2] + m[3,4] + 3 \times 6 \times 5$

$= 150 + 60 + 90$

$= 300$

$K=3$

$(A_0 A_1 A_2 A_3) A_4$

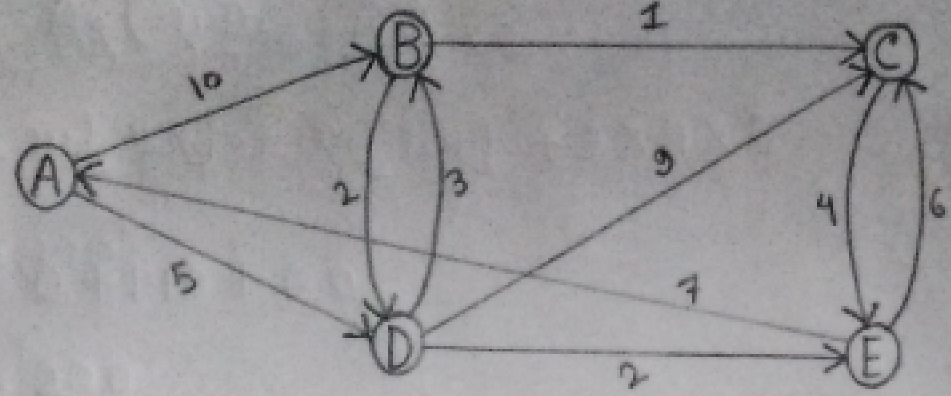
$= m[0,3] + m[4,4] + 3 \times 2 \times 5$

$= 124 + 0 + 30$

$= 154$

Shortest Path

Dijkstra's Algorithm



	A	B	C	D	E
A	0_a	10_a	α_a	5_a	α_a
D	0_a	8_d	14_d	5_a	7_d
E	0_a	8_d	13_e	5_a	7_d
B	0_a	8_d	9_b	5_a	7_d
C	0_a	8_d	9_b	5_a	7_d

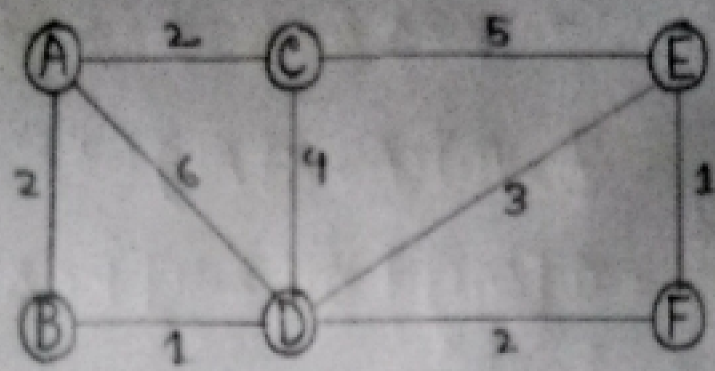
Source → A

$A \longrightarrow B = 8$

$A \longrightarrow C = 9$ Shortest Path

$A \longrightarrow D = 5$

$A \longrightarrow E = 7$



	A	B	C	D	E	F
A	0_a	2_a	2_a	6_a	α_a	α_a
B	0_a	2_a	2_a	3_b	α_a	α_a
C	0_a	2_a	2_a	3_b	7_c	α_a
D	0_a	2_a	2_a	3_b	6_d	5_d
F	0_a	2_a	2_a	3_b	6_d	5_d
E	0_a	2_a	2_a	3_b	6_d	5_d

$A \to B = 2$

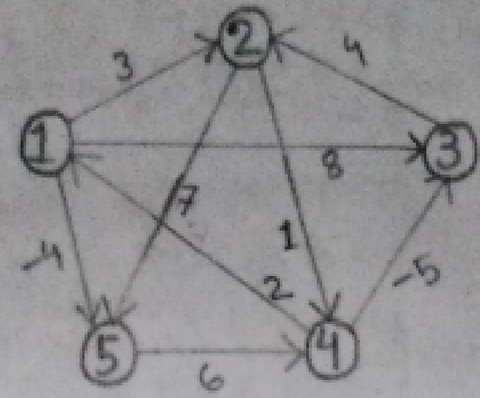
$A \to C = 2$

$A \to D = 3$

$A \to E = 6$

$A \to F = 5$

Floyd-Warshall Algorithm



[D0]

	1	2	3	4	5
1	0	3	8	α	-4
2	α	0	α	1	7
3	α	4	0	α	α
4	2	α	-5	0	α
5	α	α	α	6	0

[D1]

	1	2	3	4	5
1	0	3	8	α	-4
2	α	0	α	1	7
3	α	4	0	α	α
4	2	5	-5	0	-2
5	α	α	α	6	0

□ − 1 − □

4 − 1 − 2 = 5

4 − 1 − 5 = -2

[D2]

	1	2	3	4	5
1	0	3	8	4	-4
2	α	0	α	1	7
3	α	4	0	5	11
4	2	5	-5	0	-2
5	α	α	α	6	0

[D3]

	1	2	3	4	5
1	0	3	8	4	-4
2	α	0	α	1	7
3	α	4	0	5	11
4	2	-1	-5	0	-2
5	α	5	α	6	0

1-3-2 = 12
4-3-2 = -1
1-3-2-4 = 13
4-3-2-5 = 6
5-4-3-2 = 5

□ − 2 − □

□ − 1 − 2 − □

□ − 2 − 1 − □

3-2-5 = 11
~~3-2-5-4 = 17~~
~~1-2-5-4 = 16~~
1-2-4 = 4

3 − 2 − 4 = 5
1 − 2 − 5 = 10
4 − 1 − 2 − 5 = 12

[D_4]

	1	2	3	4	5
1	0	3	-1	4	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	8	5	1	6	0

3-2-4-1-5 = 3
2-4-1-5 = -1
2-4-1 = 3
5-4-1 = 8
2-4-3 = -4
1-2-4-3 = -1
1-5-4-3 = 4 ×
3-2-4-1 = 7
5-4-3-2 = 5
5-4-1-3 = 16 ×
5-4-3 = 1

[D_5]

	1	2	3	4	5
1	0	1	-3	2	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	8	5	1	6	0

2-5-4 = 13
1-5-4 = 2
1-5-4-3 = -3
1-5-4-3-2 = 1
2-4-1-5 = -1
3-2-5-4 = 17

This is the shortest path

What is P, NP, NP hard, NP complete? How can you prove a problem NP hard?

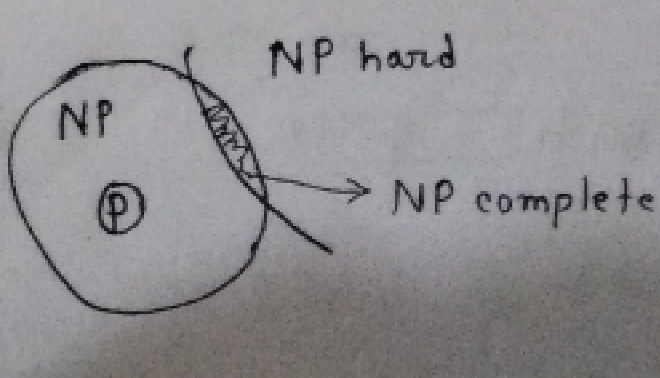
P - Those problems which can be solved in polynomial time are known as P

NP - A problem which can be solved in polynomial time by a nondeterministic machine is known as NP Problem.

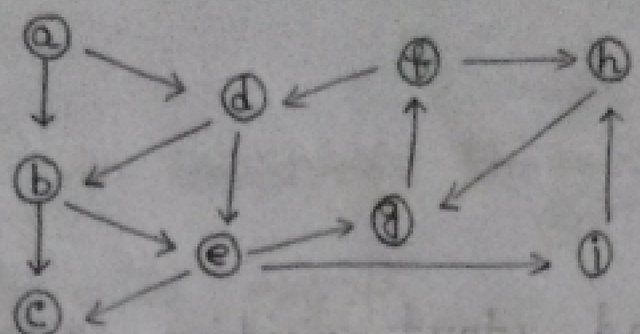
NP-hard - A problem is said to be a NP-hard problem if it has no solution in polynomial time.

NP-complete - A problem is called a NP-complete if it can be solved when nondeterministic machine is invented

Let us guess X a problem. We would like to prove X a NP-hard problem. We need to find some problems which is already proved NP-hard problem. Then the known NP-hard problem should have to be reduced in my problem and need to show that the problem can be solved if and only if my problem can be solved. So my problem is greater than that problem. As that problem is NP-hard problem so my one is also a NP-hard problem.



BFS example



[queue]

I) initialize a queue and start from a and mark it visited.

[a]

II) now need to find unvisited adjacent node from a and enque them and dequeue ~~b~~ a

[b d]

III) find unvisited node from b and enqueue them and dequeue b

[d c e]

IV) unvisited adjacent node from d and enqueue them and dequeue d

[ce]

V) for c

[e]

VI) for e

[g i]

VII) for g

[i f]

VIII) for i

f h

IX) for f

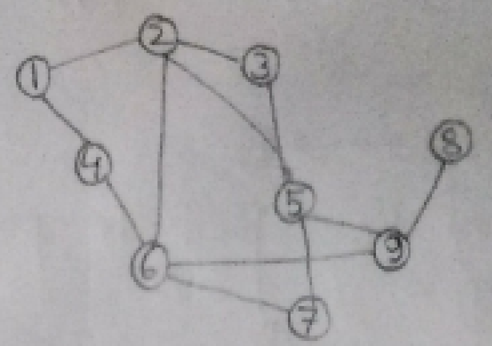
h

X) for h

blank

a b e d e f g h i j k

DFS example



I) initialize a stack and start visiting from 1. Marking it visited push on stack

$\begin{bmatrix}1\end{bmatrix}$

II) find unvisited adjacent node from 1. here 2 and 4 are unvisited. We take 2 and push

$\begin{bmatrix}2\\1\end{bmatrix}$

III) 3, 6, 5 are unvisited from 2. We take 3 and push

$\begin{bmatrix}3\\2\\1\end{bmatrix}$

IV) 5 is unvisited from 3. take and push $\begin{bmatrix}5\\3\\2\\1\end{bmatrix}$

V) 7 and 9 are unvisited from 5. take 7 and push $\begin{bmatrix}7\\5\\3\\2\\1\end{bmatrix}$

V) 6 is unvisited from 7. take and push $\begin{bmatrix}6\\7\\5\\3\\2\\1\end{bmatrix}$

VI) 4 and 9 are unvisited from 6. take 4 and push $\begin{bmatrix}4\\6\\7\\5\\3\\2\\1\end{bmatrix}$

VII) no unvisited node from 4. So pop 4 $\begin{bmatrix}6\\7\\5\\3\\2\\1\end{bmatrix}$

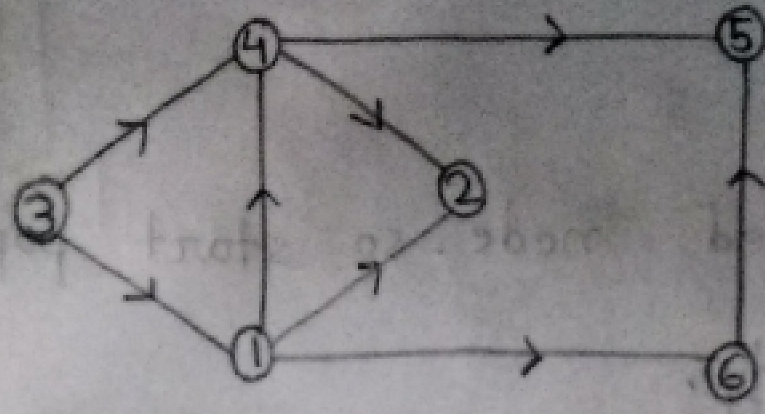
VIII) 9 is unvisited from 6. so push $\begin{bmatrix}9\\6\\7\\5\\3\\2\\1\end{bmatrix}$

ix) 8 is unvisited from 9. so push

8
9
6
7
5
3
2
1

x) so there is no unvisited node. so start popping until the stack becomes empty.

Topological sort



indegree সবচেয়ে কম অথবা শূন্য

→ 3, 1, 4, 2, 6, 5