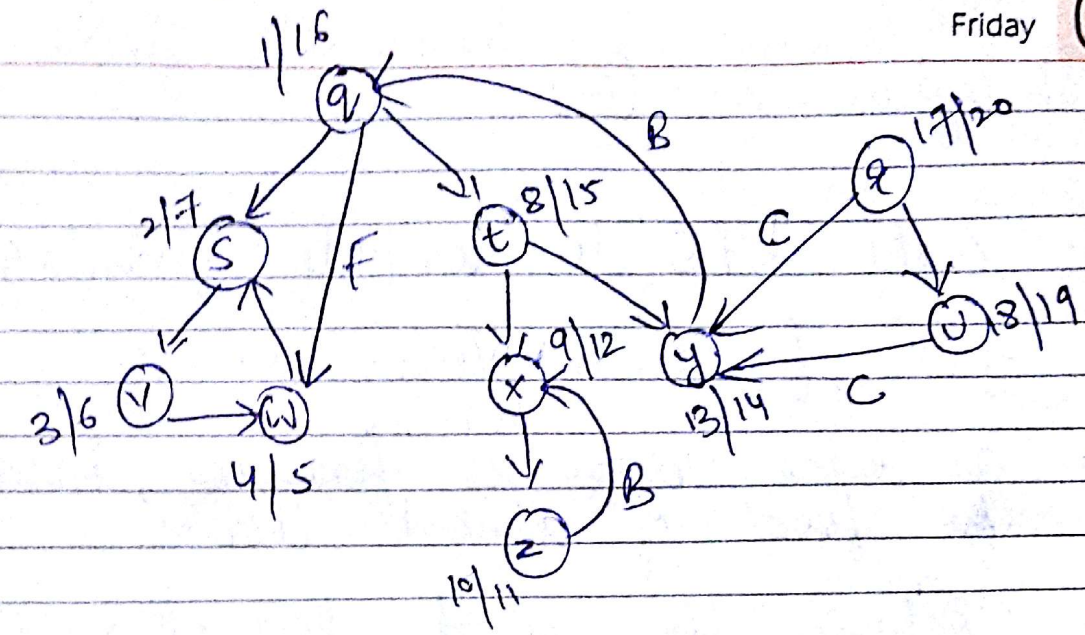
Unit-3

AUGUST 2013

Wk 31 • 214-151

Friday 02



Topological sort :- Directed acyclic graph or DAG are used for topological sorts.

Wk 31 • 215-150

Saturday 03

- A topological sort of a directed acyclic graph $G = (V, E)$ is linear ordering of $u, v \in V$ such that if $(u, v) \in E$ then u appears before v in this ordering. If G is cyclic, no such ordering exist.

Sunday 04

AUGUST 2013

05 Monday

Wk 32 • 217-148

Topological-Sort (G)

1. Call DFS to Compute finishing time $f[v]$ for each vertex v.
2. as each vertex is finished, insert it onto the front of linked list.
3. Return the linked list of vertices.

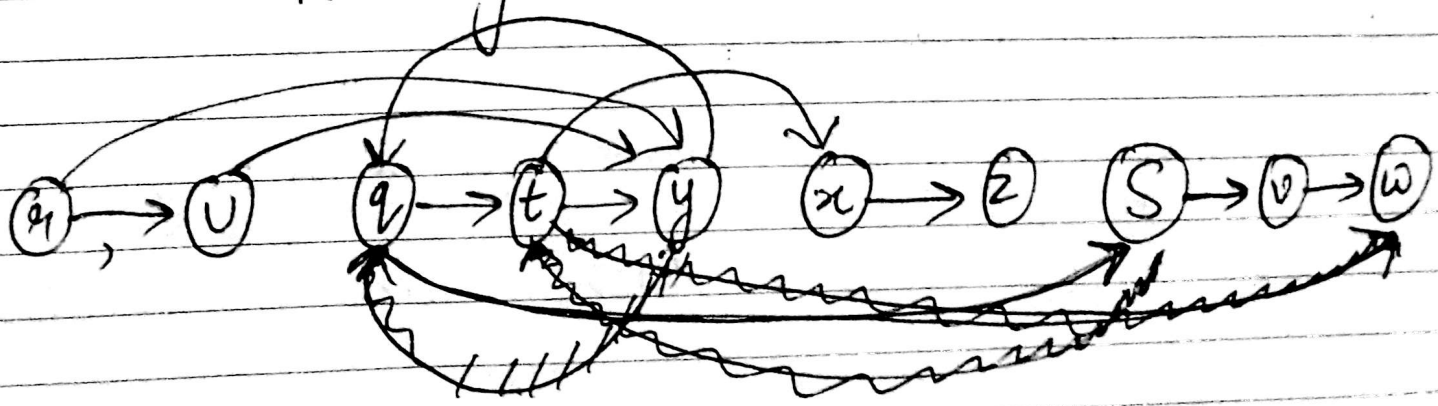
- It can Be Viewed as placing all the vertices along Horizontal line so that all directed edges go from left to right.

06 Tuesday

Wk 32 • 218-147

- DAG's are Used in many applications to indicate precedence.

Running time = $\Theta(V+E)$



Wk 32 • 219-146 Wednesday 07

Strongly connected Components

A directed graph is called strongly connected if for every pair of vertices u and v there is a path from u to v and path from v to u. The strongly connected components of a directed graph are its maximal strongly connected subgraphs.

Wk 32 • 220-145 Thursday 08

Kosaraju's algo :-

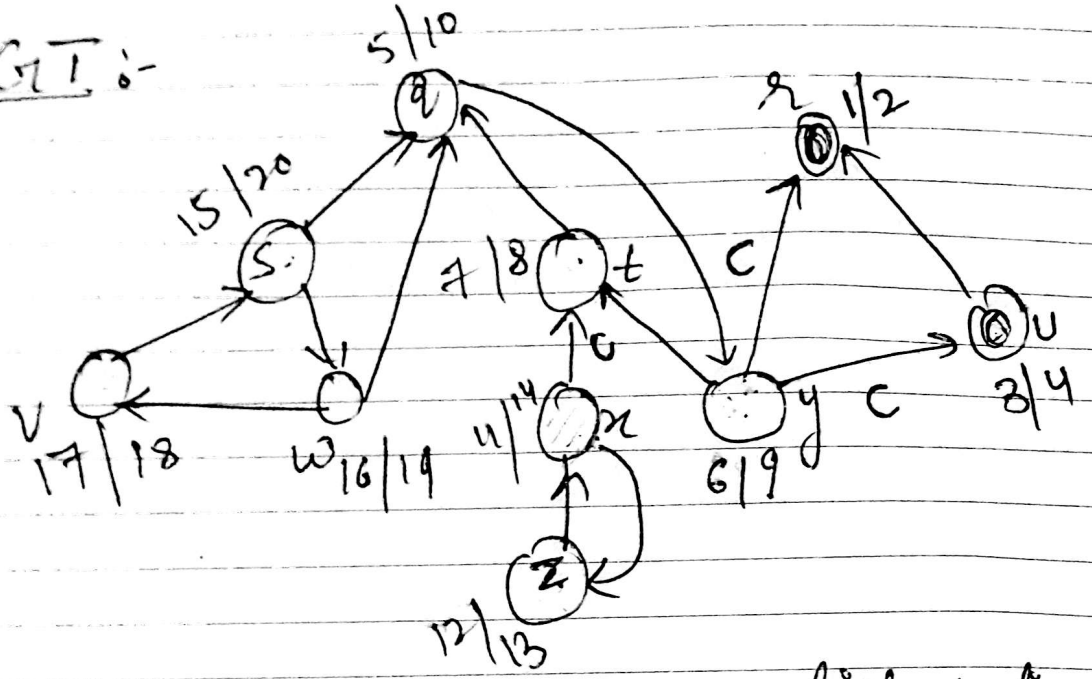
(1) Call DFS to compute finishing time $f[u]$ for each vertex u. Calculate (G^T).

(2) Call DFS (G^T) but in main loop of DFS Consider the vertices in order of decreasing $f[u]$.

(3) Produce as Output the vertices of each tree in DFS forest formed in lines as seperate SCC.

09 Friday Reverse the G1 |-

G1 :-



starting se as se as highest finishing time in G1.

10 Saturday

{s w v} {q, y, t} {x, z} {r} {u}

11 Sunday

Single Source Shortest Path

- We define the shortest path weight from u to v by $\delta(u,v) = \min(w(p) : u \rightsquigarrow v)$ if there is a path from vertex u to v and $\delta(u,v) = \infty$, otherwise.

Single source shortest paths problem :-

- Given a graph $G(V,E)$ we want to find shortest path from given source vertex $s \in V$ to every $v \in V$.

* If there is a negative cycle on some path from s to v we define $\delta(s,v) = -\infty$

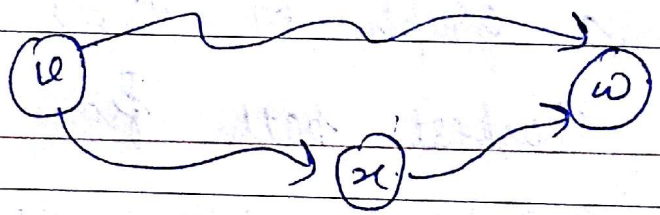
14 Wednesday

Shortest path property :-

Optimal Substructure :- All sub paths of shortest paths are shortest path.

- Triangle Inequality :- Let $d(v, w)$ be length of shortest path from v to w then,

$$d(v, w) \leq d(v, x) + d(x, w)$$



15 Thursday

Relaxation :- Single source shortest path is based on technique known as relaxation

- Is a method that repeatedly decrease an upper bound on actual shortest path weight of each vertex until upper bound equals to shortest path weight.

- for each vertex $v \in V$, we maintain an attribute $d[v]$ which is upper Bound on weight of shortest path from source s to v.

Initialize-single-source (G, s)

1. for each vertex $v \in V[G]$
2. do $d[v] \leftarrow \infty$
3. $\pi[v] \leftarrow NIL$
4. $d[s] \leftarrow 0$

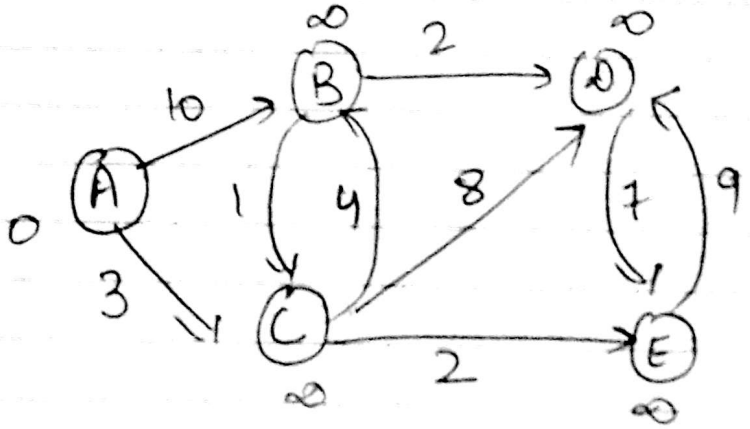
Relax :- (u, v, w)

1. if $d[v] > d[u] + w(u, v)$
2. then $d[v] \leftarrow d[u] + w(u, v)$
3. $\pi[v] \leftarrow u$

Dijkstras (G, W, S)

1. Initialize single source (G, S)
2. S ← ϕ
3. Q ← V[G]
4. While Q ≠ ϕ
5. do U ← Extract-Min (Q)
6. S ← S U {U}
7. for each vertex V ∈ Adj[U]
8. do Relax (U, V, W)

Example :-



Let A Be source vertex :-

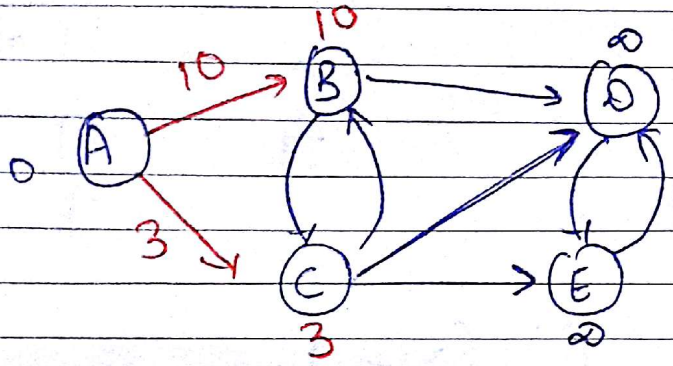
Q :-

A	B	C	D	E
0	∞	∞	∞	∞

"A" → Extract_Min(Q)

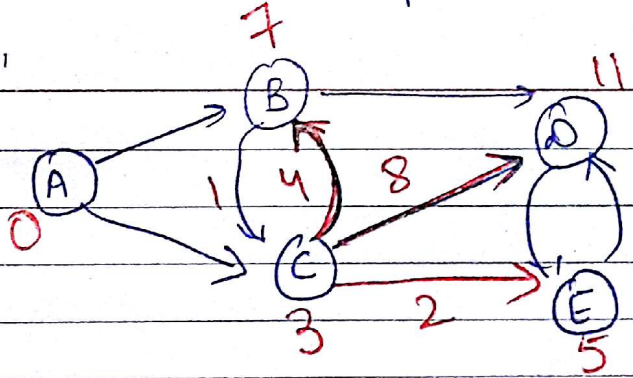
A	B	C	D	E
~~0~~	∞	∞	∞	∞

Relax All edges leaving A



A	B	C	D	E
Q: 0	∞	∞	∞	∞
	10	~~3~~	-	-

from C i-



A	B	C	D	E
0	∞	∞	∞	∞
	10	~~3~~	-	-
~~7~~	~~11~~			~~5~~

{A, C, E}

Relax all edges leaving ~~E~~ i-

A	B	C	D	E
0	∞	∞	∞	∞
	10	3	-	-
	7		11	5
			~~11~~	

SEPTEMBER 2013

M	T	W	T	F	S	S	M	T	W	T	F	S	S
						1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30						

"A great social success is a pretty girl who plays her cards as carefully as if she were plain

23 Friday

Relax all edges leaving B:-

{A, C, E, B, D}

A	B	C	D	E
0	∞	∞	∞	∞
	10	3	-	-
	7	-	11	5
	7		11	
			9	

$O(V^2+E)$

Running time = $O(V^2)$.

24 Saturday

25 Sunday

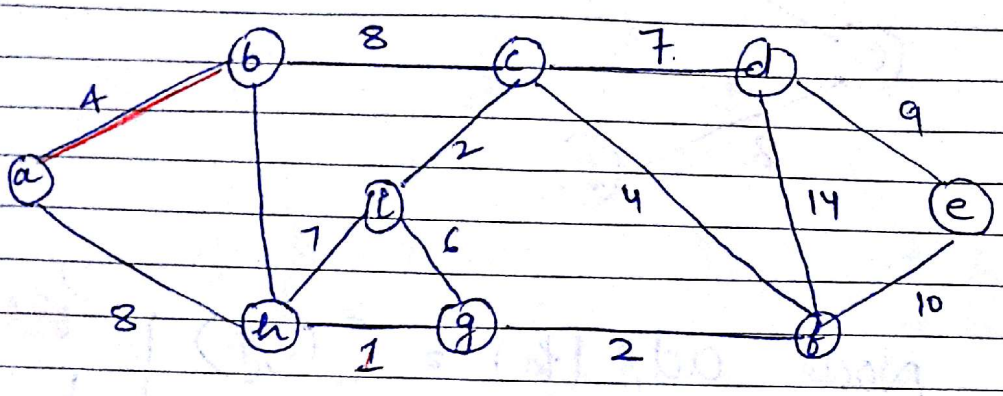
AUGUST 2013

Monday 26

Prim's Algo :-

Que

Example :-



Start from node a :- a = 0

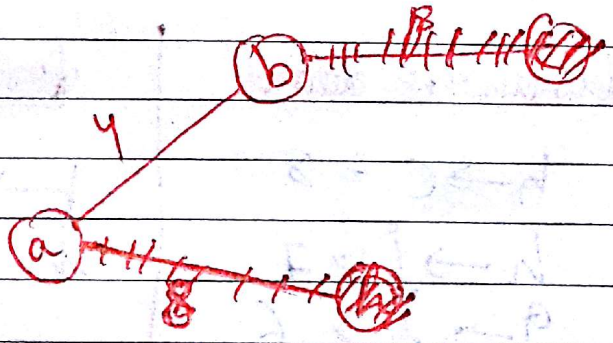
Tuesday 27

Now :- adj a = [b, h]

a → b = 4

a → h = 8

a → b = 4 is min so remove b.



from b :- adj [b] = (c, h)

a → h = 8

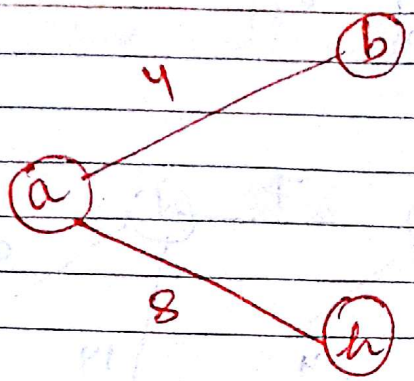
b → c = 8

b → h = 11

AUGUST 2013

28 Wednesday

Wk 35 • 240-125



Now adj [h] = (i, g)

b → c = 8

h → i = 7

h → g = 1

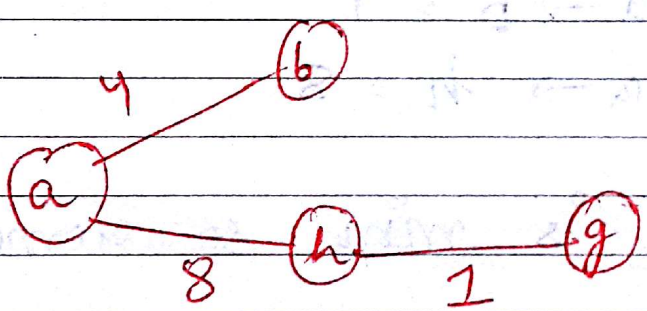
forming loops

b → h

h → b

29 Thursday

Wk 35 • 241-124



Nodes in loop

a

b

h

g

distances added

b → c = 8

h → i = 7

g → i = 6

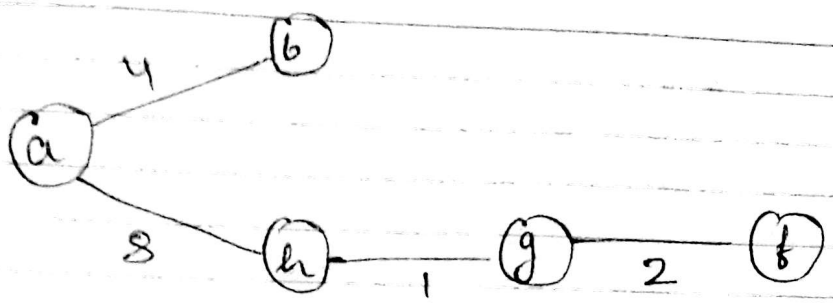
g → f = 2

forming loops

b → h

h → b

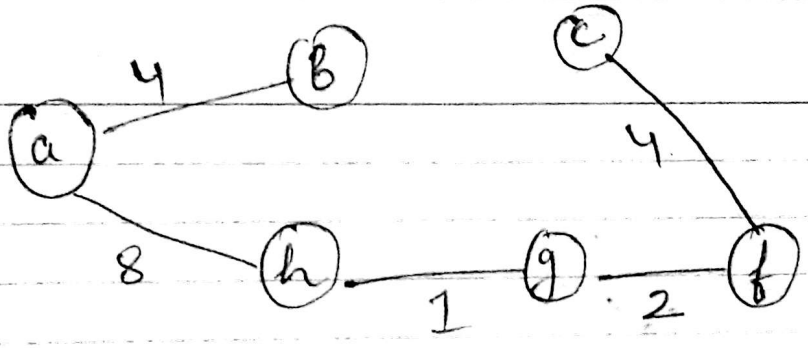
Success is getting what you want; happiness is wanting what you get



a
b
h
g
f

$b \to c = 8$
$h \to i = 7$
$g \to i = 6$
$f \to c = 4$
$f \to d = 14$
$f \to e = 10$

$b \to h$
$h \to b$



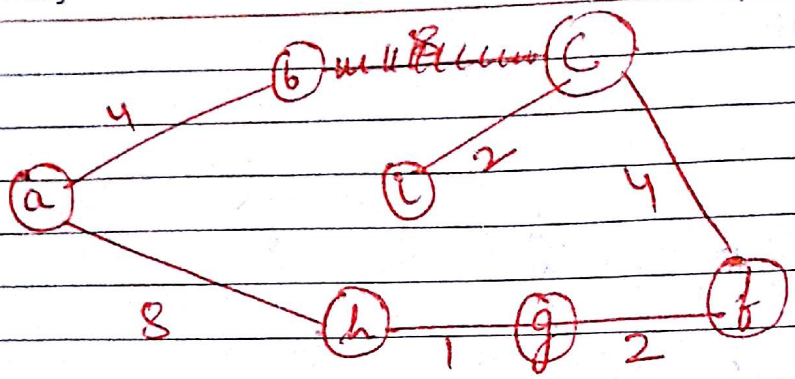
a
b
h
g
f
c

$h \to i = 7$
$g \to i = 6$
$f \to d = 14$
$f \to e = 10$
$c \to i = 2$
$c \to d = 7$

$b \to c$
$b \to h$
$h \to b$

$c \to b$

02 Monday

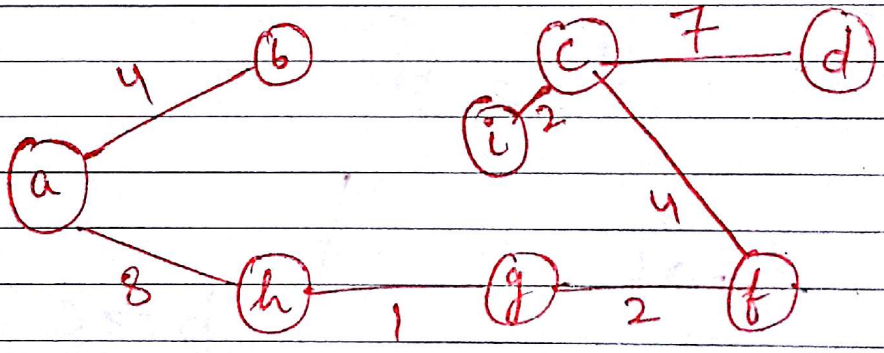


a
b
h
g
f
c
i
d

$f \to d = 14$
$f \to e = 10$
$\boxed{c \to d = 7}$

$b \to c$, $b \to h$
$h \to b$, $h \to i$
$g \to i$
$c \to b$
$i \to h$, $i \to g$

03 Tuesday



a
b
h
g
f
c
i
d

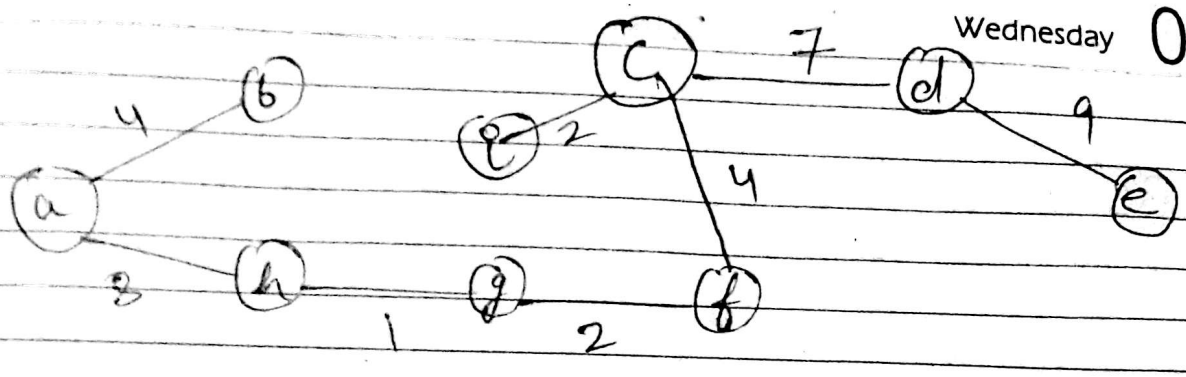
$f \to e = 10$
$\boxed{d \to e = 9}$

$b \to c$, $b \to h$
$h \to b$, $h \to i$

SEPTEMBER 2013

M	T	W	T	F	S	S	M	T	W	T	F	S	S
						1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30						

Don't think of it as failure. Think of it as time-released success



MST :- Weight = Sum of all edges

= 37 Ans

Runing Time :- O(E log V)

- Algo from given notes.
- Kruskal's from given notes.

SEPTEMBER 2013

06 Friday

Bellman ford :- $O(VE)$

Wk 36 • 249-116

- Bellman-ford Algo finds shortest path for general case in which edge weights can be negative.

- This Algo returns a Boolean value indicating whether or not there is a -ve cycle that is reachable from source. If there is such cycle the Algo indicates "no soln".

07 Saturday

Wk 36 • 250-115

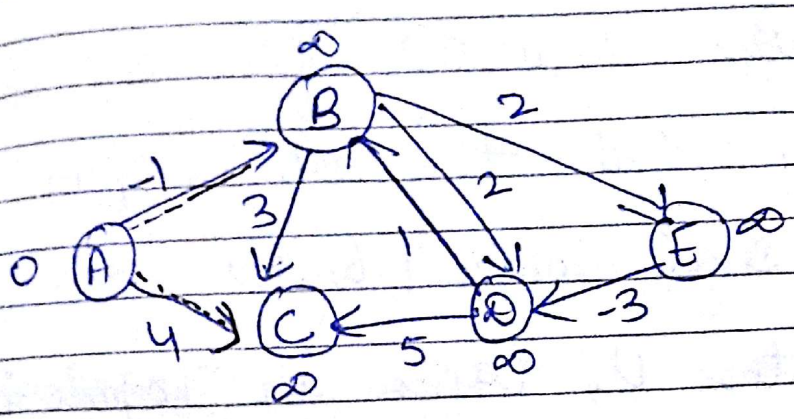
Bellman-ford (G, w, s)

1. Initialize-Single Source (G, s) } $O(V)$
2. for $i \leftarrow 1$ to $|V[G]| - 1$
3. do for each edge $(u, v) \in E[G]$ } $O(E)$
4. do Relax (u, v, w) }
5. for each edge $(u, v) \in E[G]$ } $O(E)$
6. do if $d[v] > d[u] + w(u, v)$ }
7. then return False.

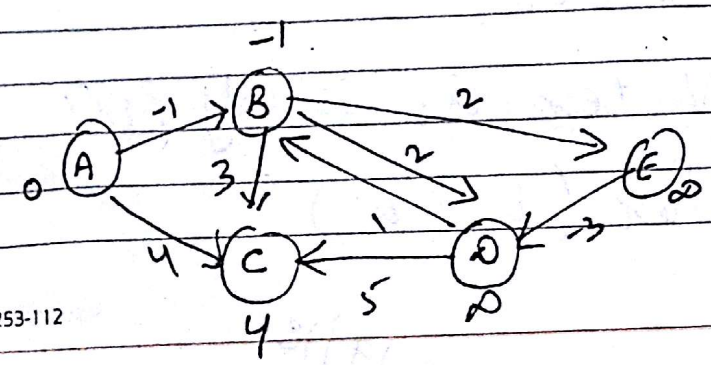
08 Sunday

Wk 37 • 252-113

8. Return True.



A	B	C	D	E
0	∞	∞	∞	∞
✓	✓	✓	✓	✓
A	B	C	D	E
0	-1	4	∞	∞
0	-1	2	1	1
0	-1	2	1	1
0	-1	2	-2	1
0	-1	2	-2	1



Wk 37 • 253-112

Tuesday 10

-1	2	-2	1
-1	2	-2	1

* This will be done 4 times Because there are 5 vertices.

(VE)

	✓ A	✓ B	C	D	E
0	0	∞	∞	∞	∞
1	0	-1	4	∞	∞
2					
3					

(OE)

OCTOBER 2013

M	T	W	T	F	S	S	M	T	W	T	F	S	S
	1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26	27
28	29	30	31										

SEPTEMBER 2013

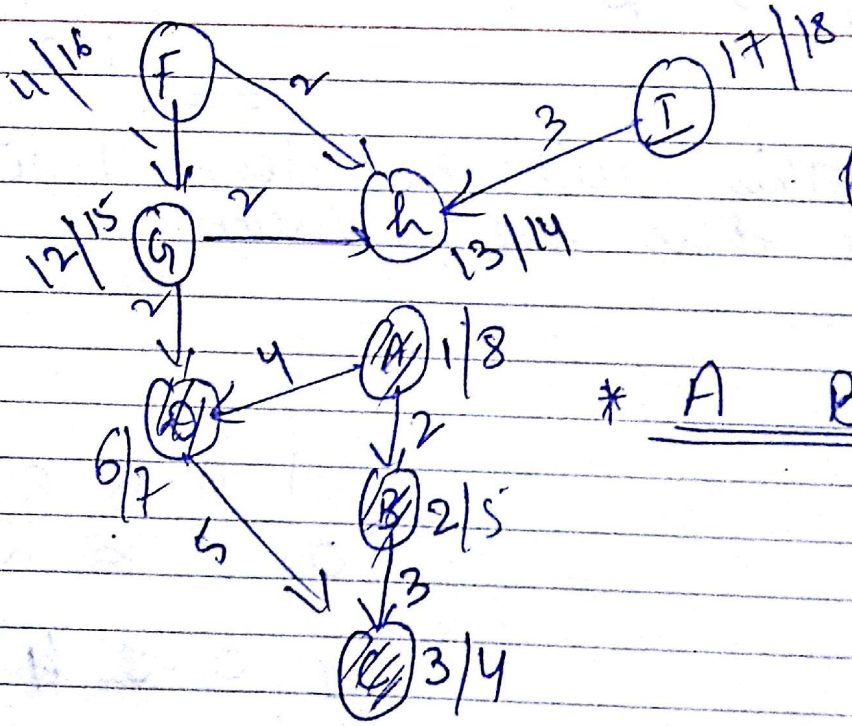
11 Wednesday

Single Source Shortest path in DAG:

DAG Shortest Paths (G, W, S)

1. Topologically sort the vertices of G
2. Initialize single source (G, S)
3. for each vertex U, taken in Topologically sorted Order
4. for each vertex $v \in G$ Adj [U]
5. Relax (U, V, W)

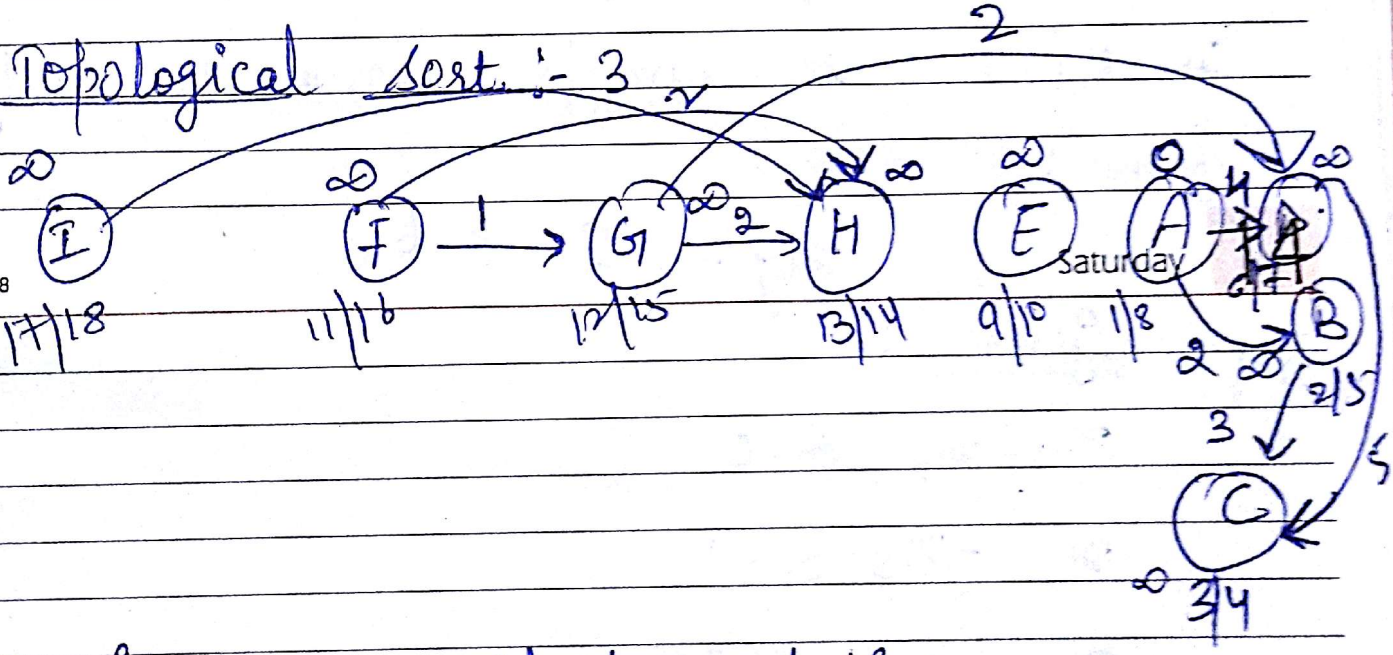
12 Thursday



* A Be S.N.

DAG :- It is directed graph with no directed cycle ie - it is a collection of vertices and directed edges in which each edge connects two vertices in such a way so that no loops or cycles are formed.

Topological sort :- 3



Single source shortest path :-

A	B	C	D	E	F	G	H
0	∞	∞	∞	∞	∞	∞	∞
			4				
	5						

16 Monday

Difference Constraint :-

- In a system of difference constraints, each row of linear - Programming matrix A contains one 1 and -1 and all other entries of A are 0. Thus, the constraints given by $Ax \leq b$ are set of m difference constraints involving n unknowns in which each constraint is simple linear inequality of form :-

17 Tuesday

$$x_j - x_i \leq b_k$$

Ques:- given :-

$$x_1 - x_2 \leq 0$$

$$x_1 - x_5 \leq -1$$

$$x_2 - x_5 \leq 1$$

$$x_3 - x_1 \leq 5$$

$$x_4 - x_1 \leq 4$$

$$x_4 - x_3 \leq -1$$

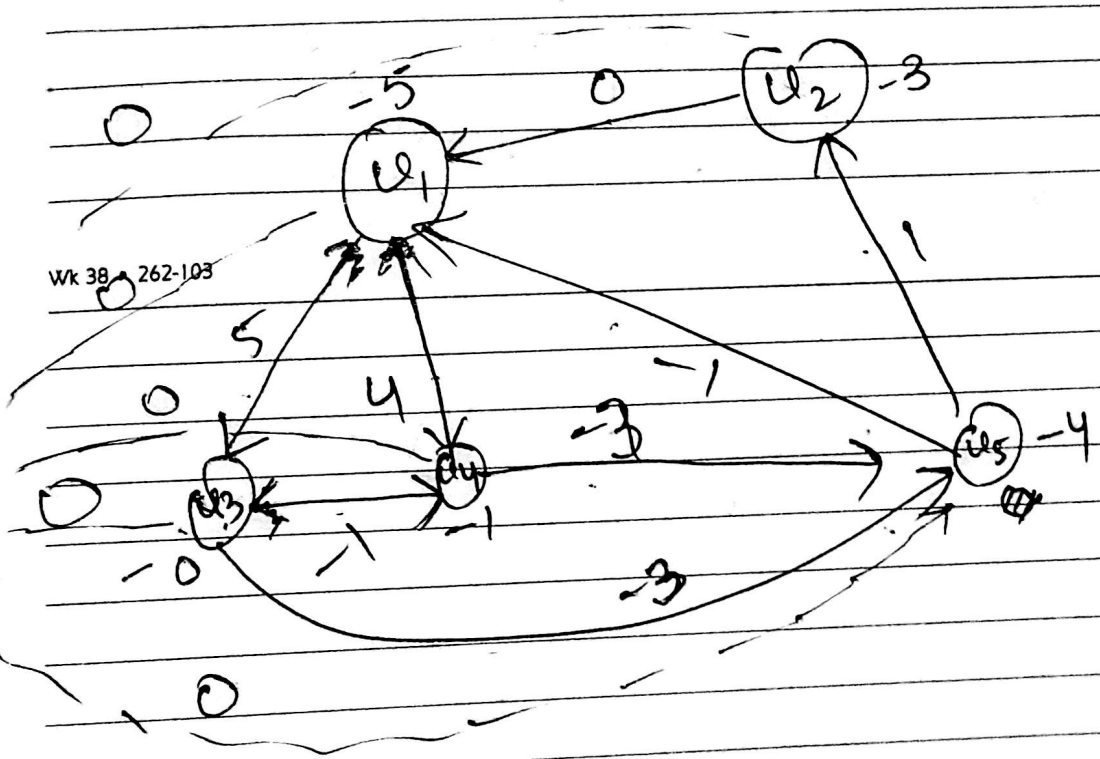
$$x_5 - x_3 \leq -3$$

$$x_5 - x_4 \leq -3$$

Wednesday 18

Solution :-

$$\begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{pmatrix} 0 \\ -1 \\ 1 \\ 5 \\ 4 \\ -1 \\ -3 \\ -3 \end{pmatrix}$$



Thursday 19

Wk 38 • 262-103

(8285

• Apply Bellman ford to solve.

SEPTEMBER 2013

20 Friday

Floyd Warshall's (All pair shortest path)

1. $n \leftarrow$ rows [W]
2. $D^{(0)} \leftarrow W$
3. for $k \leftarrow 1$ to n
4. do for $i \leftarrow 1$ to n
5. do for $j \leftarrow 1$ to n
6. do $d_{ij}^{(k)} \leftarrow \min(d_{ij}^{(k-1)}, d_{ij}^{(k-1)} + d_{kj}^{(k-1)})$
7. ↳ Return $D^{(n)}$

21 Saturday

Time Complexity = $O(V^3)$

$\pi_{ij}^{(k)}$

22 Sunday

$$d_{ij}^{(k)} = \begin{cases} \min\left[d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right] \end{cases}$$

Monday 23

Wk 39 • 266-099

$D^{(0)} =$

0	∞	∞	∞	-1	∞
1	0	∞	2	∞	∞
∞	2	0	∞	∞	-∞
-4	∞	∞	0	3	∞
∞	7	∞	∞	0	∞
∞	5	10	∞	∞	0

Tuesday 24

Wk 39 • 267-098

k	i	j	$d_{ij}^{(k-1)}$	$d_{ik}^{(k-1)}$	$d_{kj}^{(k-1)}$	$d_{ij}^{(k)}$
1	1	1	0	0	0	0
1	1	2	∞	0	∞	∞
1	1	3	∞	0	∞	∞
1	1	4	∞	0	∞	∞
1	1	5	-1	0	-1	-1
1	1	6	∞	0	∞	∞

OCTOBER 2013

M	T	W	T	F	S	S	M	T	W	T	F	S	S
	1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26	27
28	29	30	31										

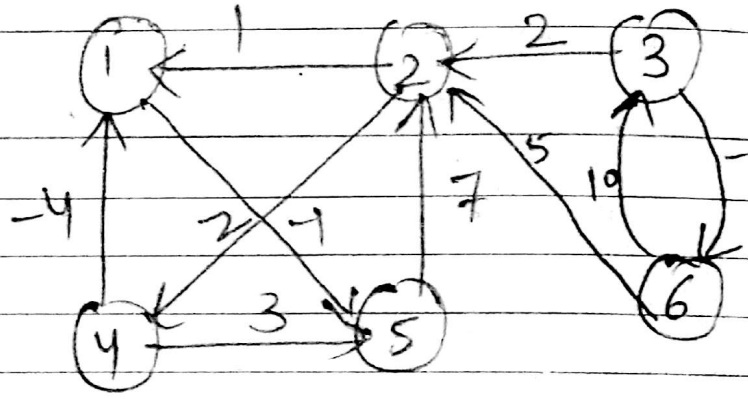
Success has always been the worst of liars

25 Wednesday

1	2	1	1	1	0	1
1	2	2	0	1	∞	0
1	2	3	∞	1	∞	∞
1	2	4	2	1	∞	2
1	2	5	∞	1	-1	0
1	2	6	∞	1	∞	∞
1	3	1				
1	3	2				

26 Thursday

1	3	3
1	3	4
1	3	5
1	3	6
1	4	1
1	4	2
1	4	3
1	4	4
1	4	5
1	4	6



SEPTEMBER 20

M	T	W	T	F	S	S	M	T	W	T	F	S	S
							1	2	3	4	5	6	7
9	10	11	12	13	14	15	16	17	18	19	20	21	2
23	24	25	26	27	28	29	30						

Success usually comes to those who are too busy to be looking for it

Friday 27

Wk 39 • 270-095

$D^{(1)} =$

$$\begin{bmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & 0 & \infty \\ \infty & 2 & 0 & \infty & \infty & 8 \\ -4 & \infty & \infty & 0 & 5 & \infty \\ \infty & 7 & \infty & \infty & 0 & \infty \\ \infty & 5 & 10 & \infty & \infty & 0 \end{bmatrix}$$

$D^{(2)} =$

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

Saturday 28

Wk 39 • 271-094

Sunday 29

OCTOBER 2013

M	T	W	T	F	S	S	M	T	W	T	F	S	S
	1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26	27
28	29	30	31										

Success is never final; failure is never fatal

30 Monday

$D^{(3)} =$

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

01 Tuesday

$D^{(4)} =$

$$\begin{bmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ 0 & 2 & 0 & 4 & -1 & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{bmatrix}$$

SEPTEMBER

M	T	W	T	F	S	S	M	T	W	T	F
						1	2	3	4	5	6
9	10	11	12	13	14	15	16	17	18	19	20
23	24	25	26	27	28	29	30				

Wk 40 • 275-090

Similarly :-

$D^{(3)} =$

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

Now we have to update parent as well :-

$$\Pi_{ij}^{(k)} = \begin{cases} \Pi_{ij}^{(k-1)} & \text{if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \\ \Pi_{kj}^{(k-1)} & \text{if } d_{ij} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \end{cases}$$

Wk 40 • 276-089

The all Pair shortest Path Problem can be Considered the mother of all Routing Problems.

- It aims to Compute the shortest Path from Vertex v to every other V.
- It is based on "dynamic Programming".

04 Friday

$\pi_{ij}^{(0)} = \begin{cases} \text{NIL} & \text{if } i = j \text{ or } w_{ij} = \infty \\ i & \text{if } i \neq j \text{ and } w_{ij} < \infty \end{cases}$

$D^{(5)}$ =

K	i	j	dij
5	1	1	
5	1	2	
5	1	3	
5	1	4	
5	1	5	
5	1	6	
5	2	1	
5	2	2	
5	2	3	
5	2	4	
5	2	5	
5	2	6	
5	3	1	
5	3	2	
5	3	3	
5	3	4	
5	3	5	
5	3	6	

05 Saturday

$D^{(6)} = D_2$

	1	2	3	4	5	6
1	0	6	∞	8	-1	∞
2	2	0	∞	2	-3	∞
3	-5	-3	0	-1	-6	-8
4	-4	-2	∞	0	-5	∞
5	7	9	∞	9	0	∞
6	3	5	10	7	2	0

Wk 40 • 278-087

Shortest Path :-

3 to 1 = -5 units

06 Sunday

Graph Unit-3 - (1)

A graph is a mathematical Model that describes such situation in the form of collection of Vertices and edges. (E, V). An edge connects two vertices.

There are two common searching algo :-

(1) BFS (Breadth first search)
(2) DFS (Depth first search)

Both of these work on directed or undirected graph.

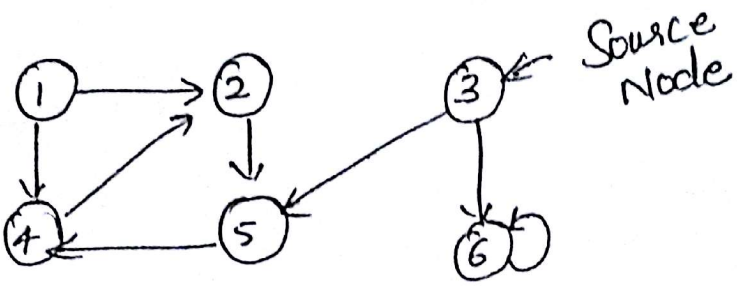
BFS :- Breadth first search have nice property of BFS tree really is a shortest path tree starting from its root. Every vertex has a path to root, with path length equal to its level and no path can skip a level so this really is a shortest path.

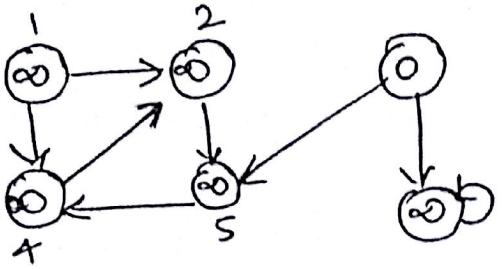
BFS (G, s)

1. For each vertex $u \in V[G] - \{s\}$
   do color [u] ← White
2. $d[u] \leftarrow \infty$
3. $\pi[u] \leftarrow$ NIL
4. Color [s] ← Gray
5. $d[s] \leftarrow 0$
6. $\pi[s] \leftarrow$ NIL
7. $Q \leftarrow \phi$
8. For
9. [illegible]

```
10      while Q ≠ ϕ
11          do u ← Dequeue (Q)
12              for each v ∈ Adj [u]
13                  do if color [v] = White
14                      then color [v] ← Gray
15                          d[v] ← d[u]+1
16                          π[v] ← u
17                          Enqueue (Q,v)
18          Color[u] ← Black
```

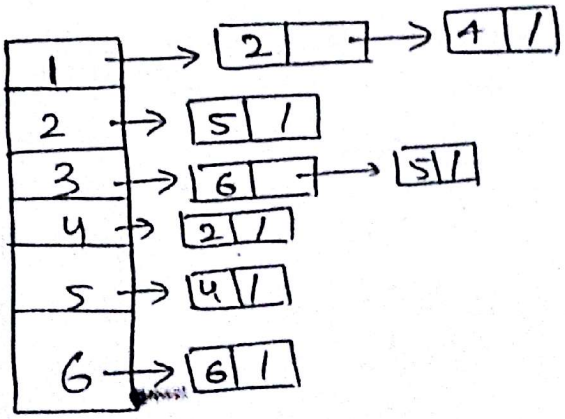
Solution :-





Q → | 3 |
0

Create adjacency list :-



adj [4] = {6, 5}

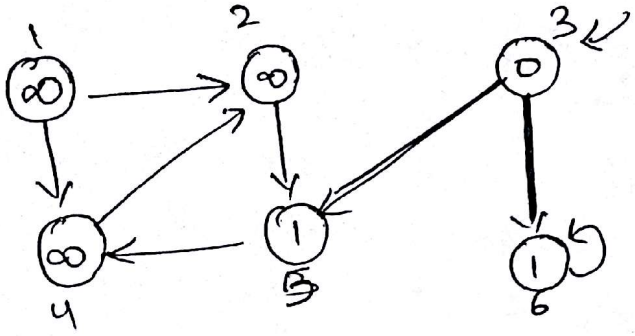
So color [6] = Gray

color [5] = Gray

d[6] = 1 d[5] = 1

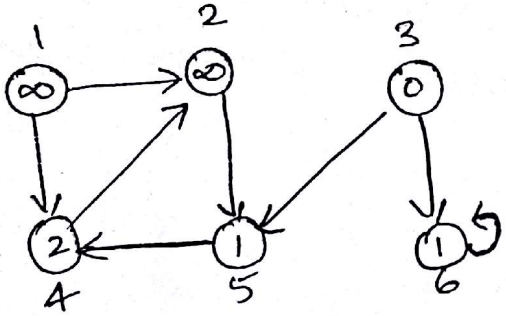
π[6] ← 3 π[5] ← 3

Example :-



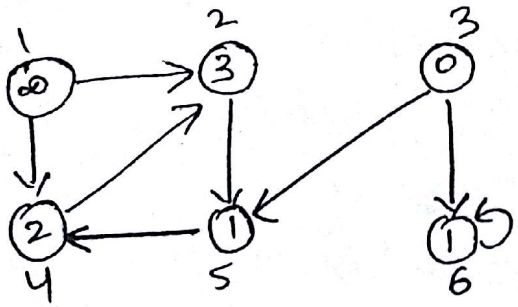
3 is source node
∴ d is 0.

Q | 5 | 6 |
1 1

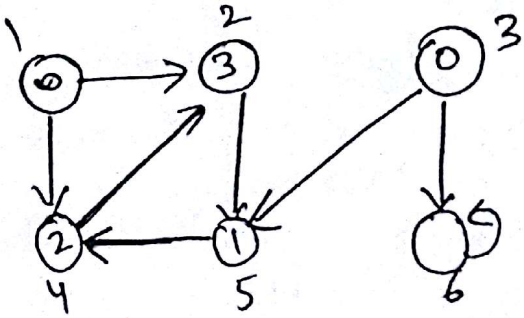


Q | 6 | 4 |
1 2

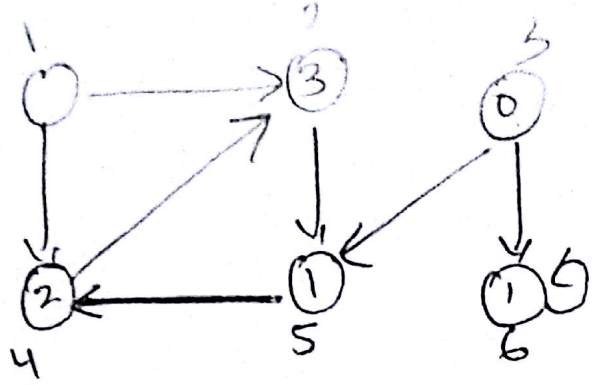
d[u] + 1 = 2



Q | 2 | |
3



Q | 2 | |
3



Q [ϕ]

DFS → Depth first search

In depth-first search, edges are explored out of the most recently discovered vertex v that still has unexplored edges leaving it. When all of v's edges have been explored. The search continues until we have discovered all vertices that are reachable from original source and it backtracks to explore edges leaving the vertex from which v was discovered.

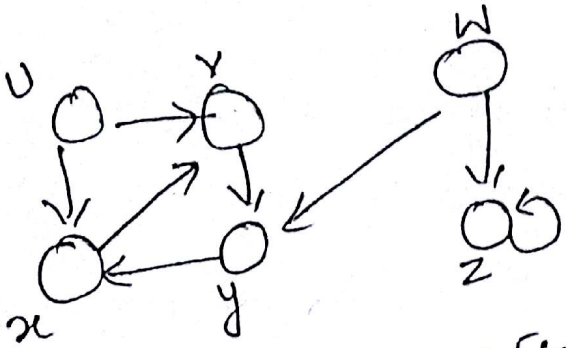
DFS visit (U)

1. Color [U] ← Gray
2. time ← time + 1
3. d[U] ← time
4. for each V ∈ adj [U]
5. do if color [V] = White
6. then π[V] ← U
7. DFS-visit (V)
8. Color [U] ← Black
9. f [U] ← time ← time + 1

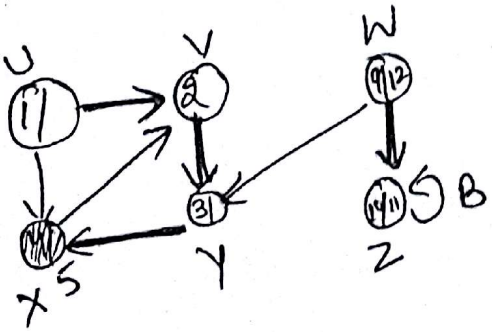
(G)

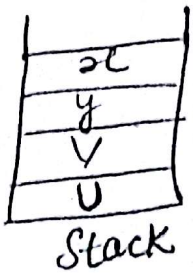
1. for each Vertex $U \in V[G]$
2. do color $[U] \leftarrow$ White
3. $\pi[U] \leftarrow$ NIL
4. time $\leftarrow 0$
5. for each Vertex $U \in V[G]$
6. do if color $[U]$ = White
7. then D-FS - Visit (U)

6.X



for each Vertex $U \in V[G]$
do color $[U]$ = White
$\pi[U]$ = NIL





adj U (V, x)

adj (V) = y

adj (y) = x

Now adj (x) = V it is already visited as it is grey now f(x) = 4+1 = 5 and make it black now Backtrack.

f(y) = 5+1 = 6

6+1 = 7

Now take node W

$d[w] = 9$

$adj[w] = z$

$d[z] = 10$

$adj(z) = z$

$f(z) = 10+1 = 11$

Backtrack :-