Final Exam

Articular point algo

Connected Component: এক node থেকে সবগুলো node এ traverse.

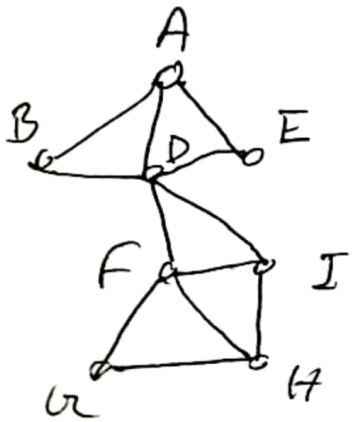
Bi-Connected: (যেখানে Path থাকে node এ)

Articulation point: যেটা remove করলে একটি graph এ ভাগ হয়ে যায়।

if there in no A.Point then it will be bi-connected.

✓ **Strategy:** ① Remove vertices
② Test DFS

- $O(n(m+n))$
- Dfs এ যখন যে edge গিয়ে এক হয় back-edge.

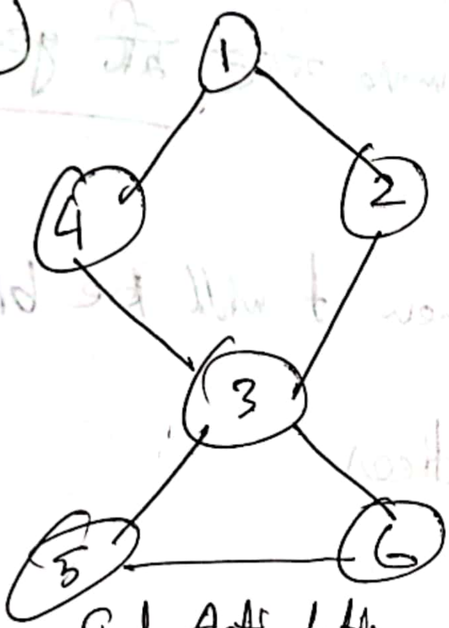


DFS ⇒

√ **low:** low = d[w] backedge
〃 low[w] tree edge

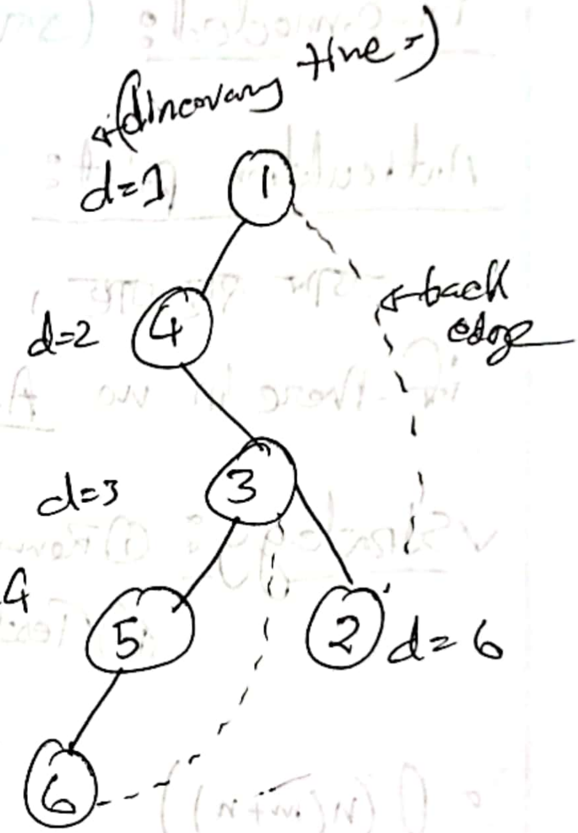
Bani (D) lt

(2)



find Articulation Point!

DFS ⇒



(1), 2, 3, 4, 5, 6

d ⇒ 1, 6, 3, 2, 4, 5

(back edge) be ⇒ 1, 1, 1, 1, 3, 3 (Path going back to parent)

Parent → U , V ← child vertices .

iff $Low[v] \geq d[u]$

the u is an articulation point except root.

$\therefore$ for 3 $\Rightarrow$ $Low[3] \geq d[4]$

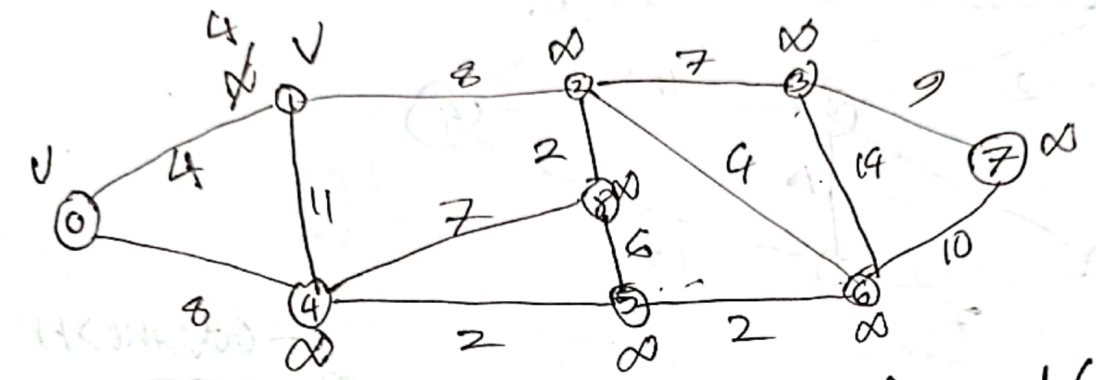
$= 1 \geq 2$ not true

for 5 $\Rightarrow$ $Low[5] \geq d[3]$

$= 3 \geq 3$ true.

$\therefore$ (3) is the articulation point.

Single Source Shortest Path Problem : Dijkstra Algo



here Source → 0

if (d(u) + c(u,v) < d(v))

⇒ 0 + 4 < ∞ , (1) → node

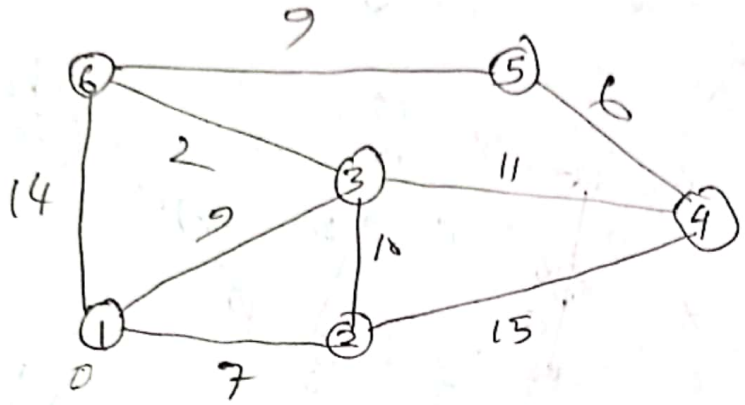
4 < ∞

then, d[v] = d(u) + c(u,v)

same way, node (4) (4) . d[v] = 8

- used for directed & Undirected.
- used in google map → shortest path.
- Greedy Method
- Relaxation : if d(u) + c(u,v) < d(v)
  d(v) = d(u) + c(u,v)

ex2



Dijkstra's

⇒

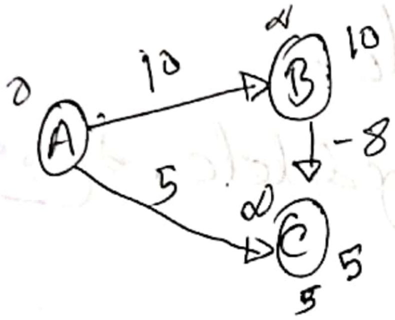
Source	Destination 2	3	4	5	6
	∞	∞	∞	∞	∞
1, 2	(7) min	9	∞	∞	14
1, 2, 3	(7)	(9)	22	∞	14
1, 2, 3, 6	(7)	(9)	(20)	∞	(11)
1, 2, 3, 6, 4	(7)	(9)	(20)	20	(11)
1, 2, 3, 6, 4, 5	(7)	(9)	(20)	(20)	(11)

Watch yours ①

Bellman Ford Algo.

Dijkstra vs Bellman ford

Dijkstra



A	B	C
0	∞	∞
AC	10	5
ACB	10	5

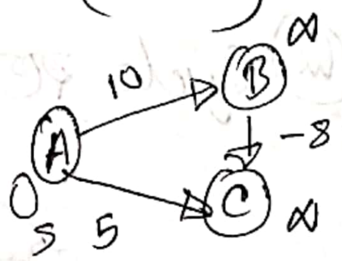
Then B & C? C = -2 হবে
আবার কিন্তু Dijkstra stop
হয়ে যায়।

Bellman ford

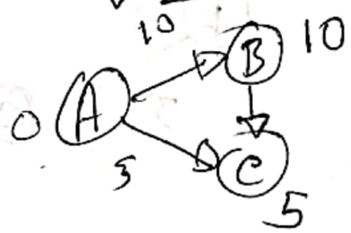
you have to relax every edge (V-1). or (twice).

(3-1) = 2

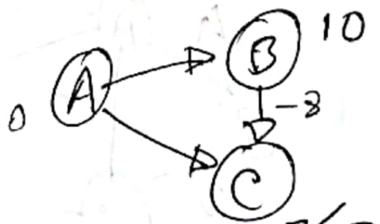
2 বার R.



R₁ (Relax 1)



R₂



~~5~~ 2 cause 2 < 5

so, Relaxation

তাহলে min ans. পাওয়া যায়।

Min Spanning Tree

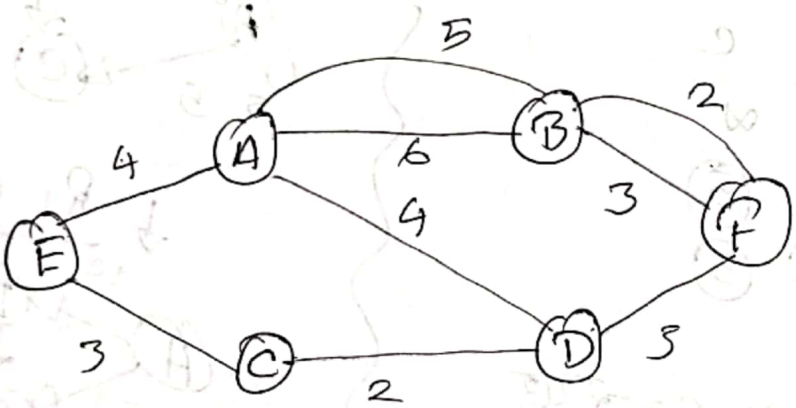
Kruskal Algo:

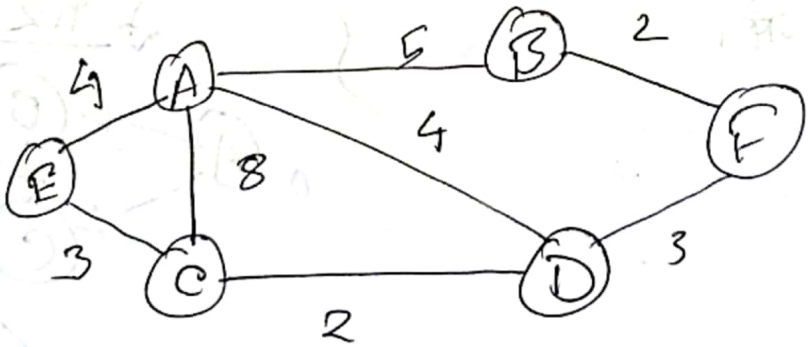
Step: (i) Graph থেকে loop delete.

(ii) Graph থেকে Parallel edge delete. ও (কম size এর)

(iii) Cycle হলে না.

Ex:

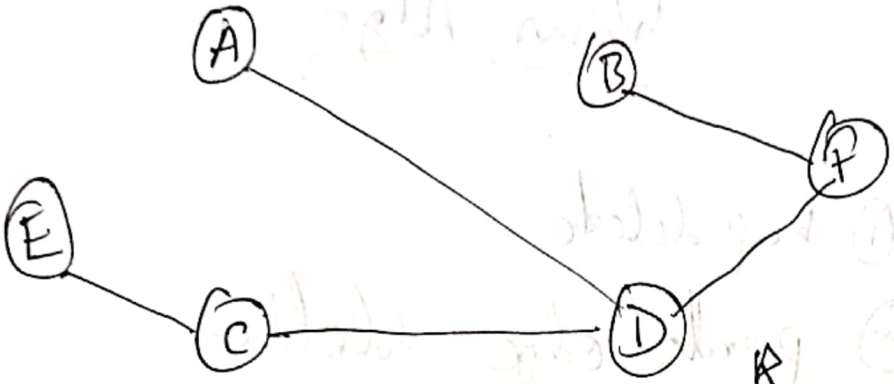




min wight edge গুলো.

BF → 2, CD → 2, DF → 3, CE → 3, AD → 4,

AB বা, AE → 7, AC → 8



Check condition for min spanning tree

① Connected, ② $(V-1)$

(SP) Spanning Tree

SP → Connected subgraph 'S' of graph $G(V, E)$ is said to be spanning iff

→ ① 'S' should contain all vertices of 'G'

② 'S' should contain $(|V|-1)$ edges.

Greedy Method

MST Rules

Condition -

$G(V, E)$, $G'(V', E')$

- $V' = V$
- $E' \subseteq E$; $E' = |V|-1$.

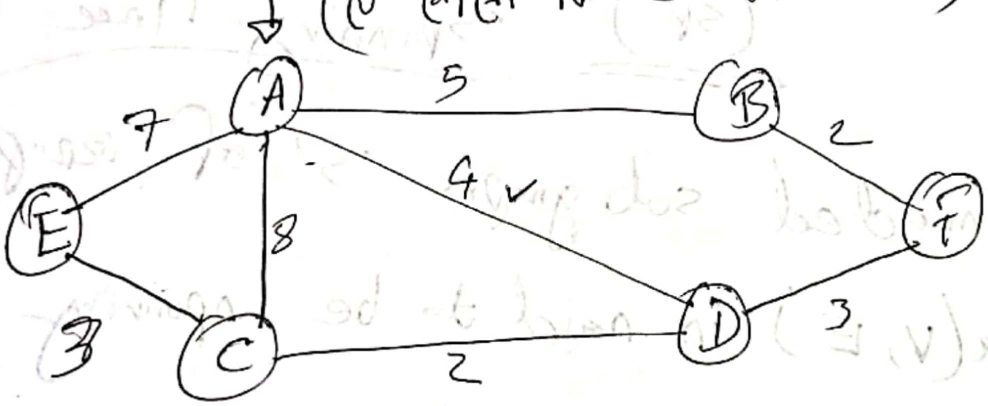
Prims Algo

Steps : (i) loop delete

(ii) parallel edge deletion

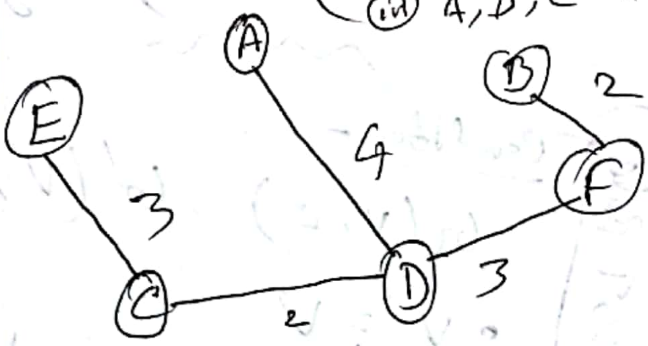
(iii) No cycle.

ex- (যে কোনো Node ধরতে হবে)



(i) A ও D উভয় থেকে min edge weight (নিয়ে যাবে)

(ii) A, D, C " -- " .



$$V' = V$$
$$E' = |V| - 1$$

Network flow / Ford Fulkerson

(Youtube)

Terms:

Source

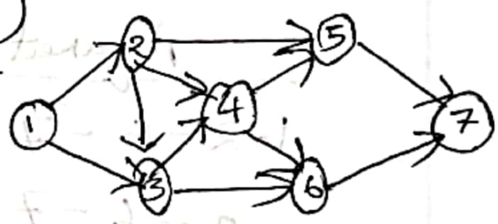
Sink

Capacity on bottle neck capacity → (min capacity of any edge on path.)

Flow (Flow → 4/5 ← capacity)

Argumenting path (Bottle neck হবে যে path দিয়ে যাব হয়।)

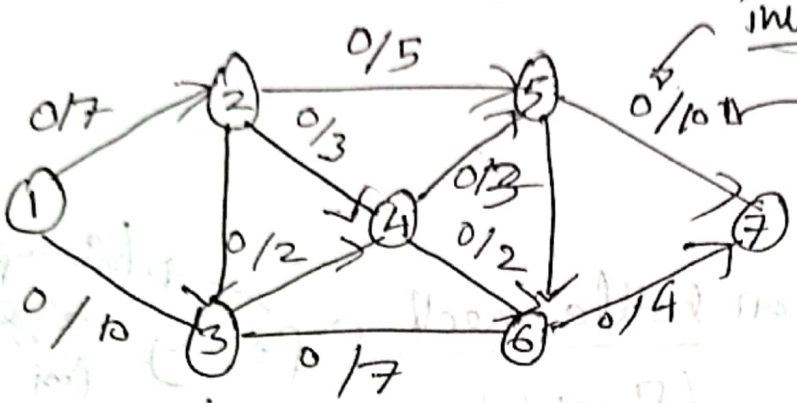
Residual capacity (original capacity − flow)



STEPS:
(i) Initially flow হবে '0' হবে।

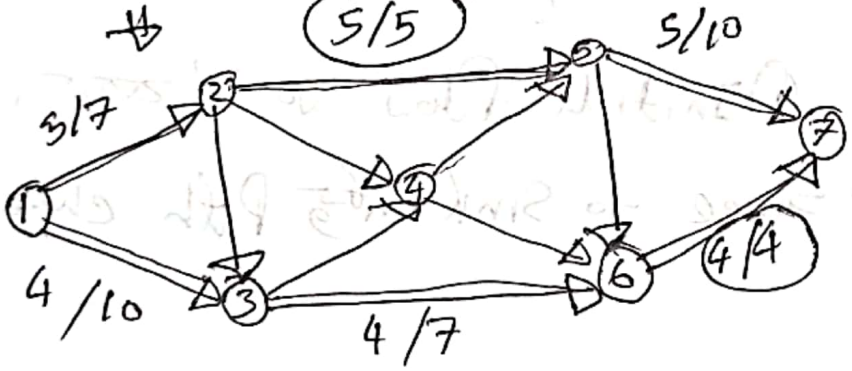
(ii) Source to sink একটা Path choose.

initial flow 0

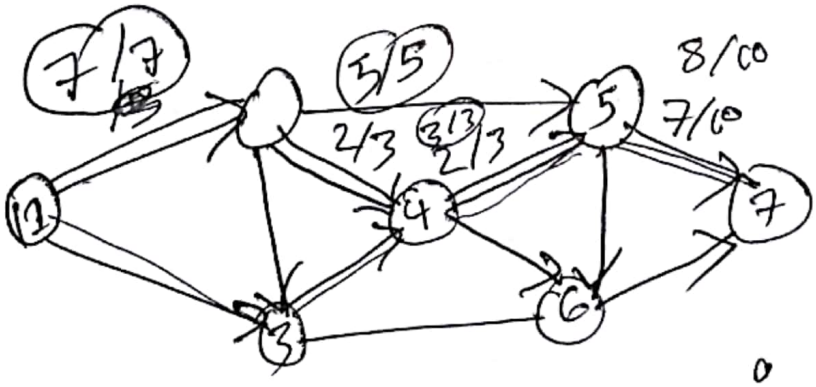


Augment Path	Bottleneck Capacity
1-2-5-7	5
1-3-6-7	4
1-2-4-5-7	2
1-3-4-5-7	1 blocked

5/5



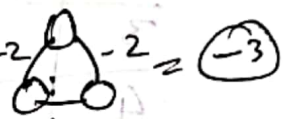
✓ Now we check Residual Capacity - (flow - capacity)



(All pair shortest path) Floyed Warshal Algo

- Dynamic approach.

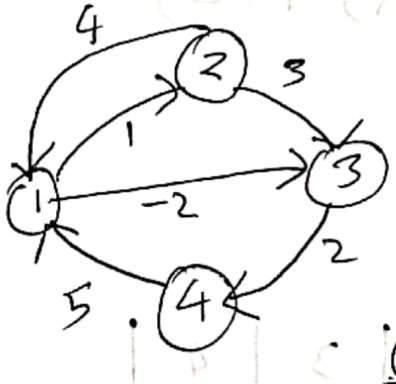
→ Works with neg & Pos edges.

but no neg cycle. ⇒ []

Formula:

$$D^k[i,j] = \min\{D^{k-1}[i,j], D^{k-1}[i,k] + D^{k-1}[k,j]\}$$

eg-



here, k=1

$D^1[3,3] = \min\{D^0[3,3], D^0[3,1] + D^0[1,3]\}$

$= \min\{3, 4+(-2)\}$

$\Rightarrow 2$

D^0 (Distance MATRIX)

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

D^1:

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

D^2

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

$$D^2[1,3] = \min\{D^1[1,3],\ D^1[1,2] + D^1[2,3]\}$$

$$= \min\{-2,\ 1+2\}$$

$$= -2$$

D^3

	1	2	3	4
1	0		-2	
2		0	2	
3	∞	∞	0	2
4				0

D^4

D^4 से result में हमें shortest path.