Final Jahir

Articular point algo

Connected component: এক node থেকে অন্যান্য node এ traverse.

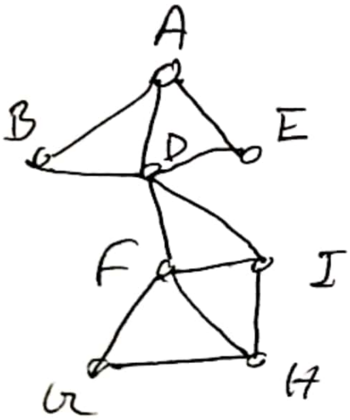
Bi-connected: (একাধিক Path থাকে node এ)

Articulation point: যাকে remove করলে দুটি graph এ ভাগ হয়ে যায়।

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

✓ **Strategy:** (i) Remove vertices
(ii) Test DFS

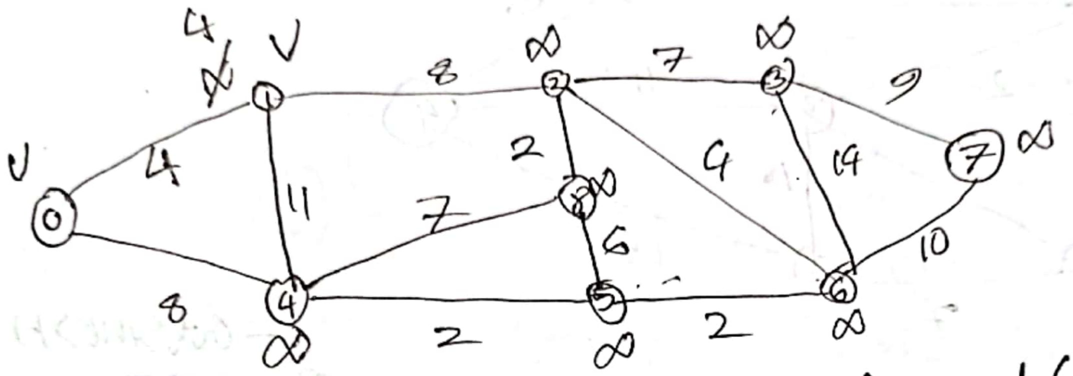
- $O(n(m+n))$
- Dfs এর সময় যে edge বাদ যায় তাই back-edge.



DFS $\Longrightarrow$

Single Source Shortest Path Problem : Dijkstra Algo

here Source → 0



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

$\Rightarrow 0 + 4 < \infty$, (1) → node

$4 < \infty$

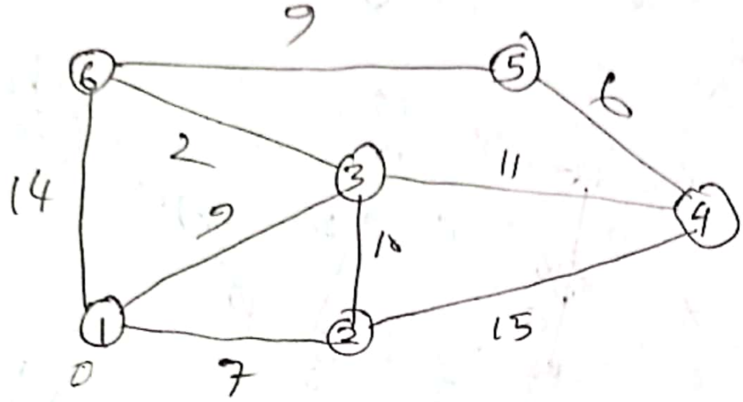
then, $d[v] = d(u) + c(u,v)$

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

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

$d(v) = d(u) + c(u,v)$

ex3



GATSMESH

=D

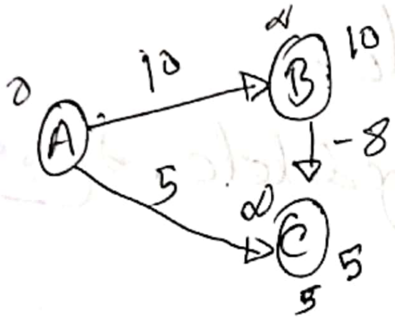
Source 1	Destination 2	3	4	5	6
	∞	∞	∞	∞	∞
1,2	7 min	9	∞	∞	14
1,2,3	7	9	22	∞	14
1,2,3,6	7	9	20	∞	11
1,2,3,6,4	7	9	20	20	11
1,2,3,6,4,5	7	9	20	20	11

Quatch pany (D)

Bellman Ford Algo.

Dijkstra vs Bellman ford

Dijkstra



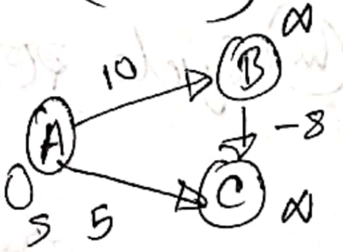
A	B	C
0	∞	∞
A C	~~10~~	5
ACB	10	5

Then B & C? C = -৩ হবে
আবার কিন্তু Dijkstra stop হয়ে যায়।

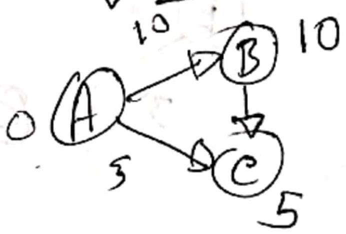
Bellman ford

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

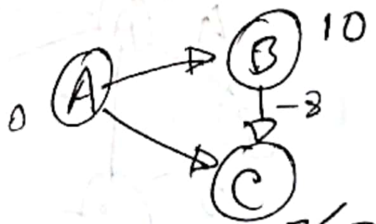
$(3-1)=2$
⇓
2 বার R.



R_1 (Relax 1)



R_2



~~5~~/2 cause $2 < 5$

so, Relaxation

তখন min ans. পাওয়া যায়।

Min Spanning Tree

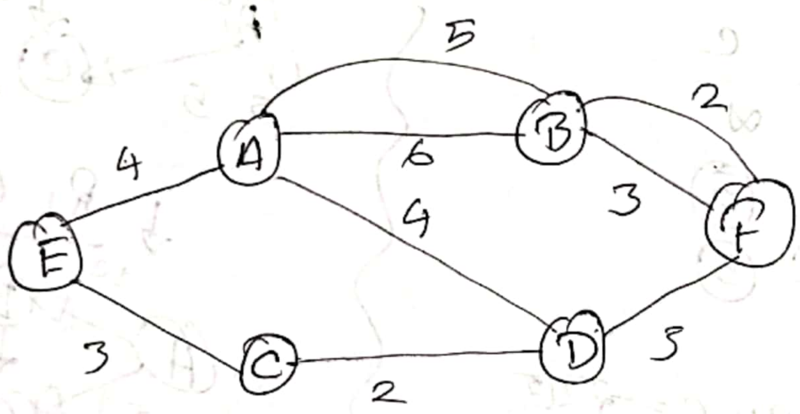
Kruskal Algo:

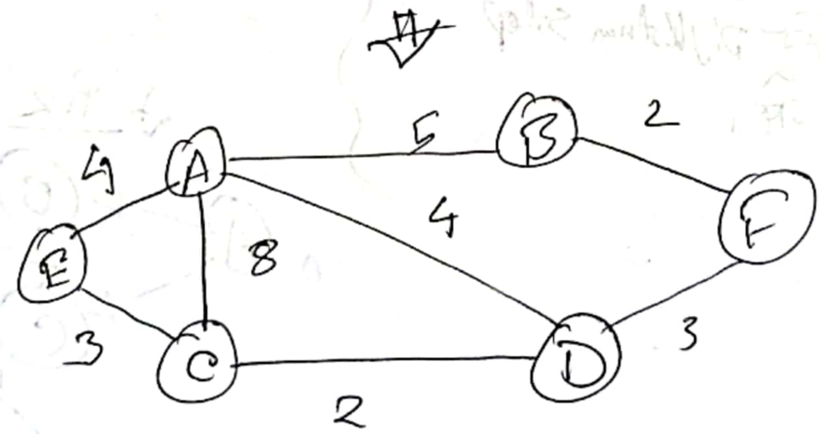
Step: ① Graph থেকে loop delete.

② Graph এর Parallel edge delete. (বড় size বাদ)

③ Cycle হবে না.

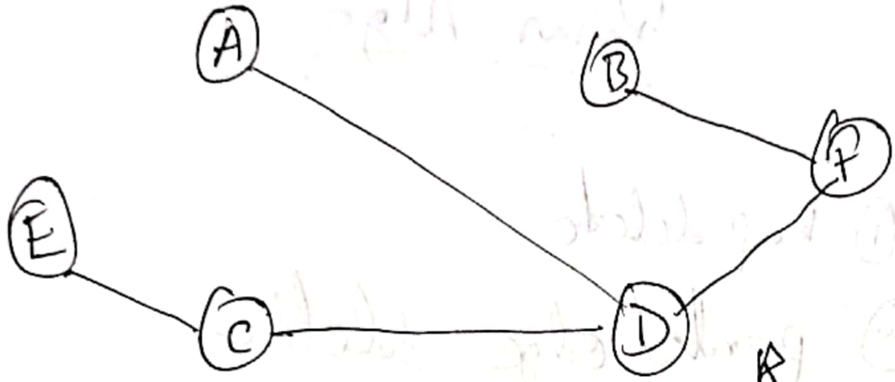
Ex:





min weight edge আগে.

$BF \to 2$, $CD \to 2$, $DF \to 3$, $CE \to 3$, $AD \to 4$,
$AB \to 5$, $AE \to 7$, $AC \to 8$



A

Check conditions for min spanning tree

(i) Connected, (ii) (V − 1)

(SP) Spanning Tree

A

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

→ (i) 's' should contain all vertices of G

(ii) 's' should contain (|V| − 1) edges.

o

w Greedy Method

MST Rules

Condition –

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

- $V' = V$
- $E' \subset E$; $E' = |V| - 1$

Prims Algo

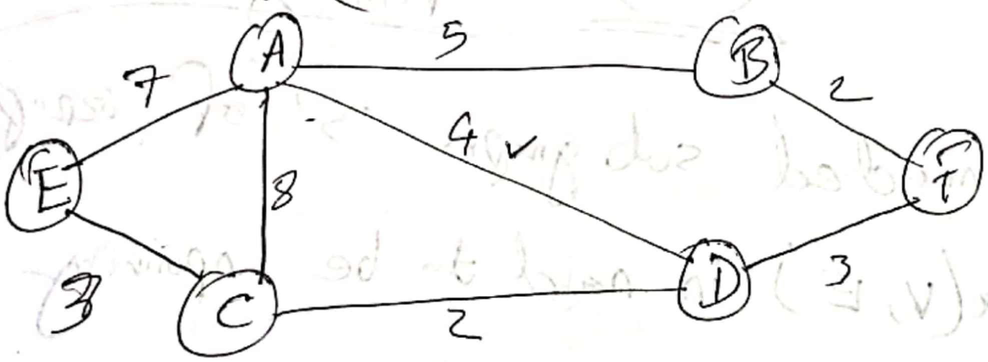
Steps : ① loop delete

② parallel edge deletion

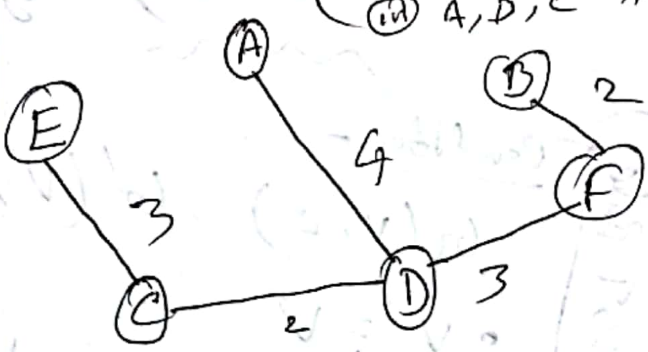
③ No cycle.

ex-

(যে কোনো Node থেকে শুরু হবে)



(① A ও D এর থেকে min edge weight (কম যেটা)
② A, D, C " - - - " .)



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

Network flow / Ford Fulkerson

(Youtube)

#Terms:

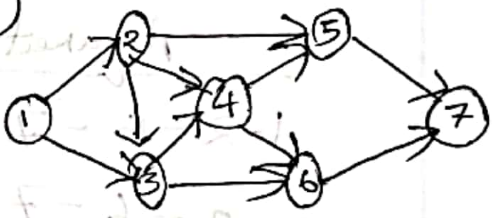
Source

Sink

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

Flow (flow → 4/5 ← capacity)

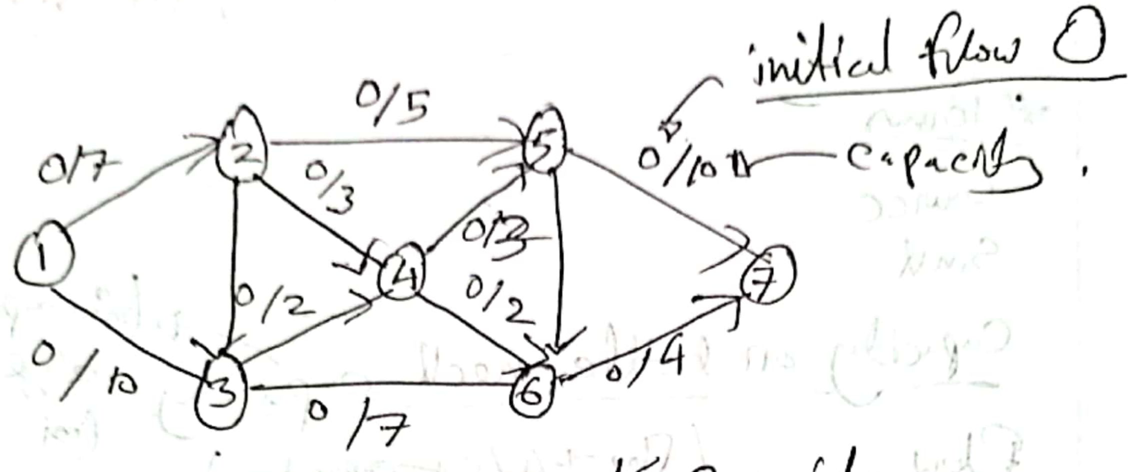
Argumenting path (Bottle neck যেক আছে যে Path দিয়ে যাওয়া হয়।)



Residual capacity (original capacity − flow)

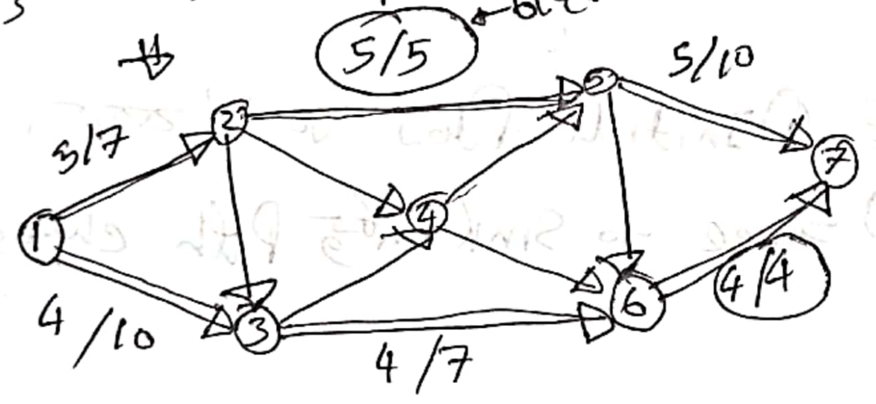
STEPS: ① Initially flow সব '0' ধরবো।

(ii) Source to Sink পর্যন্ত Path choose.



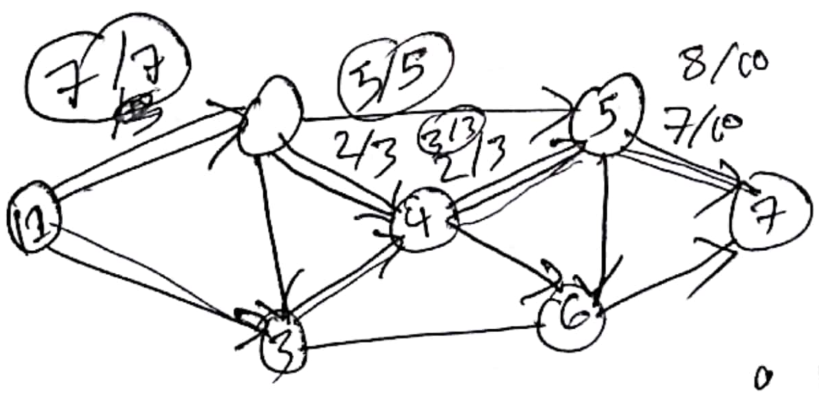
Augment Path	Bottleneck Capacity
1-2-5-7	5
1-3-6-7	4
1-2-4-5-7	2
1-3-4-5-7	1

blocked



flow = 5 + 4

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



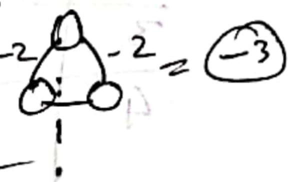
flow = 5 + 4 + 2 + 1

Max flow = 12

(All pair shortest path) Floyed Warshal Algo

- Dynamic approach.
- → Works with neg & Pos edges.

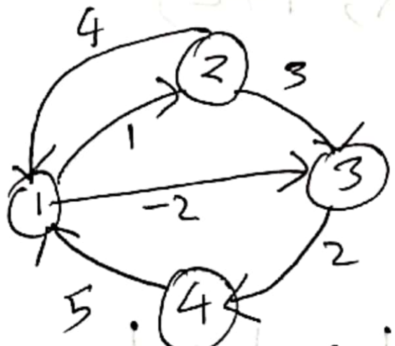
but no neg cycle. ⇒



Formula:

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

ex-



Here, D^1, $k=1$

$D^1[2,3] = \min\{D^0[2,3],\ D^0[2,1] + D^0[1,3]\}$

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

$\Rightarrow 2$

D^0 (Distance MATRIX)

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

① [illegible]

D^1:

	1	2	3	4
1	0	1	-2	∞
2	4	0	3	∞
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 यह result है हमारा shortest Path.