" Network Design OT "

Network design in supply chain is a strategic approach to planning and optimizing a company's supply chain structure.

Supply chain Network Design & Optimization (SCNDO) is a process that helps companies improve their supply chain using data analysis and planning methods.

* Network problems examples
1. Gas pipe connections
2. Water pipelines
3. Infrastructure Projects
4. Book writing Projects
5. Transportation problem
6. Metro stations
7. Planar Graphs
8. Social Network
9. Computer Network

Network Problems : 2 types

a) Network Routing Problem
b) Network Scheduling Problem

* Network Routing Problems
- shortest route algo.
- Minimal spanning trees
- Seven Bridge prbl
- Maximal flow Algo.
- Minimal cut flow
- Euler / Hamiltonion
Graph theory

- Sensor networks
- Algo. that address to social networking issues, etc.

* Network Scheduling Problem / Project Management related.
↳ (time, cost, completion) schedule
- Infrastructure project
- write research project, etc.

eg. Infrastructure Project.
↳ Activities of the project
1. Clear the site
2. Survey and layout
3. Rough Grade
4. Excavate for electrical manhole, etc.

List of scheduling Project
a) Identify activities
b) Identify activities dependencies
c) Identify network to execute the project
d) Identify resource optimization

To do above
↳ we start representing project as a network comprised of activities and analyze the network to understand the time / cost optimization.

Project management techniques
a) CPM (Critical Path method) - Deterministic model
b) PERT (Program Evaluation and Review tech.) - Probabilistic model

c) RAMP (Review Analysis of multiple project)

Draw network in Project Management

A Network in content of Project management is a graphic representation of a project's operations and is composed of activities.

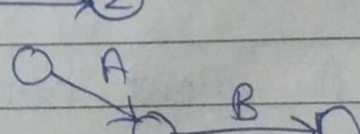
eg.

Activities	Predecessor Activity	Duration (days)
A	-	5
B	-	3
C	-	4
D	-	7
E	A, B	3
F	E	5
G	F	2
H	D	2
I	G, H	3
J	C, I	7

Basic components of network

1) Activity - task / job
2) Event / Node - task started at ① & finish at ②.
   It has start time & end time & arrow type to show direction.

   ① ⟶ ②



3) Preceding Activity -
4) Sucessor activity -

Error

1) Loop - (Network are not comprised of loop)
2) Dangling process - Incomplete join of an activity.
   A project start from one node & finish at one node.
   - two endings are not allowed. or starting

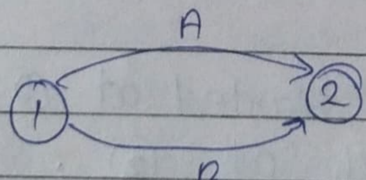
Rules of Network Construction

1. Each activity representing by one & only one arrow.

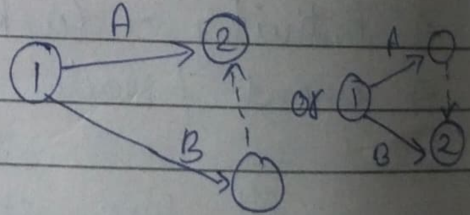
2. Each activity must be identified by starting & ending node.

3. Nodes should be labelled.

4. B/w any pair of nodes, there should be only one activity.
   OR two nodes cannot have simultaneous start or simultaneous end.



not allowed



5. Arrows should be straight.

6. - An event cannot occur until all incoming activities into it have been

completed.

- An activity cannot start unless all the preceding activities on which it depends, have been completed.

- Dummy activities should only be introduced if absolutely necessary.

eg. ① Project management.

A & B - starting activities

A, B precede C

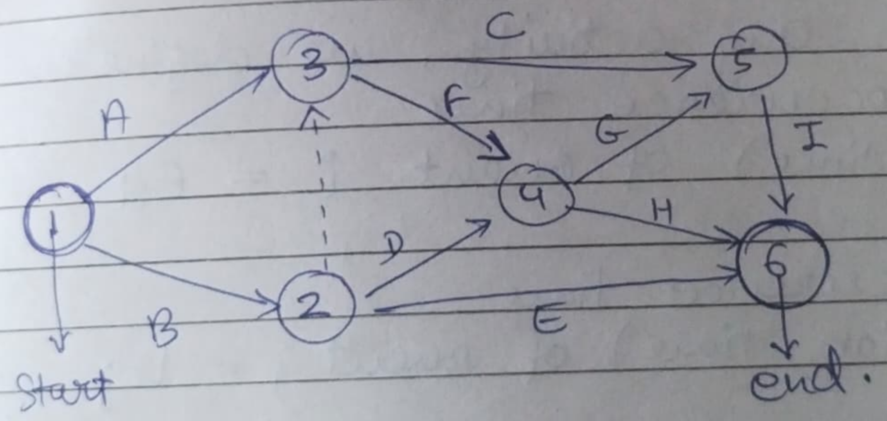
B precede D & E

A & B precede F

F & D precede G & H

C & G precede I

E, H & I are terminal activities.



eg. ② Project

A - starting

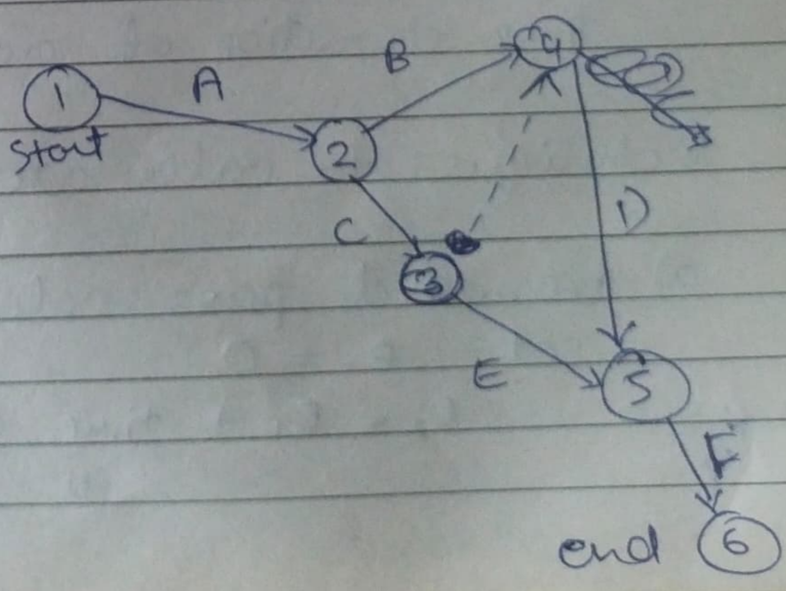
A precedes B & C

C precede D & E

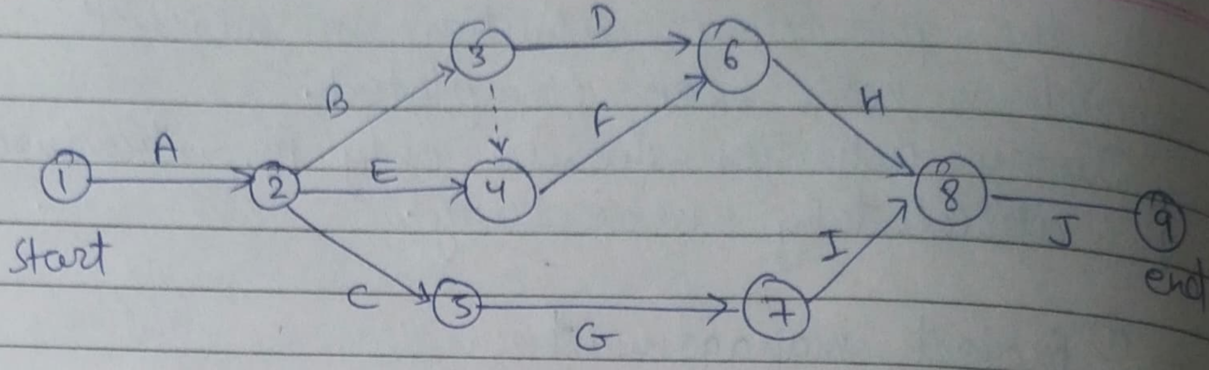
F follows E

D precedes of F

B precedes of D.



Eg ③ Activities : A B C D E F G H I J
Precedors : - A A B A B,E C D,F G H,I



Critical Path Analysis

- Total duration (completion time of Project)
- Identify critical or non critical activities
- Identify critical path

* Total duration

define : For any activity, we define :

1) Earliest occurence time (starting time) of event i = E_{si}

2) Latest occurence time (completion time) of event j = L_{cj}

time duration of activity $(i, j) = t_{ij}$

techniques to calculate E_{si}

a) Forward pass calculation (E_{si})

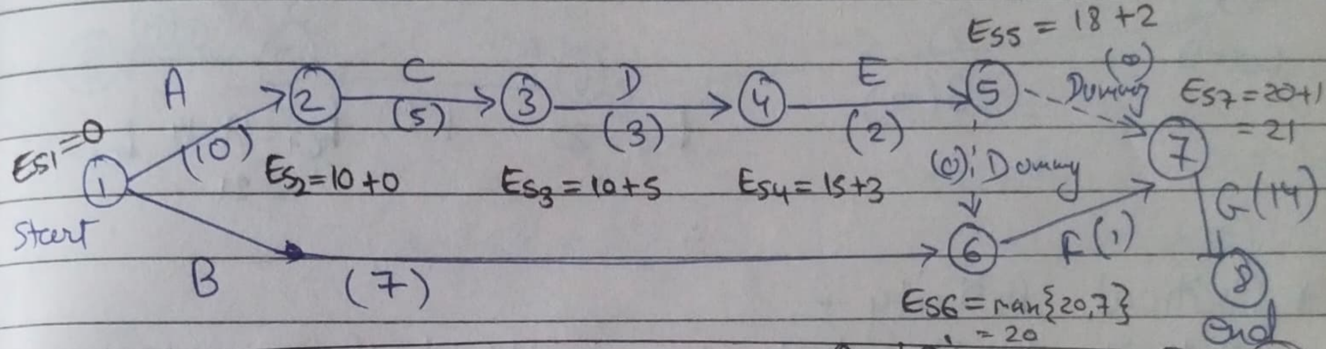
set $E_1 = 0$

$E_2 = E_1$ + time take by next activity (all events) - choose max.

$$\therefore \boxed{E_{si} = E_i = \max\{E_{i-1} + t_{ij}\}}$$

eg.

A	-	10
B	-	7
C	A	5
D	C	3
E	D	2
F	B, E	1
G	E, F	14



Notation : (E_{si}, L_{cj}) or [box: L_{cj} above, E_{si} below] or $E_{si} =$ $L_{cj} =$

technique to calculate L_{cj}

a) Backwark Pass calculation

set $L_l = E_l$; ↳ subscript stand for last node.

and to compute latest occurence times of event i $(i < j)$;

subtract duration of each activity from lastest finish time of activity.

that is $\boxed{L_i = \min\{L_j - t_{ij}\}}$

* forward & Backward Pass calculation ensures

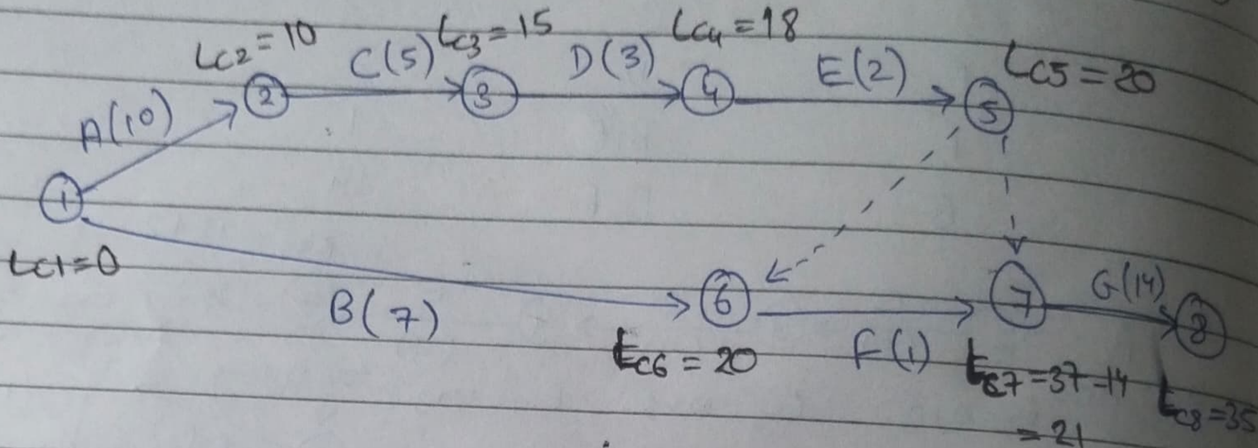
1) $E_1 = L_1$ 1 → first node

2) $E_l = L_l$ l – last node

eg.

Act.	A	B	C	D	E	F	G
Prec.	–	–	A	C	D	B, E	E, F
duration	10	7	5	3	2	1	14

$L_{c8} = E_{s\ell} = L_{c\ell} = 35$



* Critical Activity –
Activity (i, j) will be critical if it satisfies following :

a) $E_{si} = L_{ci}$
b) $E_{sj} = L_{cj}$
c) $E_{sj} - E_{si} = L_{cj} - L_{ci} = t_{ij}$

If any one of above condition is not satisfied then it is <u>non critical activity</u>.

\# Critical Path : Path which comprises only critical activities. spans entire network is called Critical path.

a) $E_{si} = L_{ci}$ ✓

b) $E_{sj} = L_{cj}$ ✓

c) $E_{sj} - E_{si} = L_{cj} - L_{ci} = t_{ij}$

↳ for B not satisfied

$① \xrightarrow{B(7)} ⑥$

$L_{c1} = 0$ $L_{c6} = 20$

$E_{s1} = 0$ $E_{s6} = 20$

$(20 - 0 \neq 7)$ ✗

→ $⑤ \overset{(0)}{\dashrightarrow} ⑥$

$E_{s5} = 20$ $E_{s6} = 20$

$L_{c5} = 20$ $L_6 = 20$

$(20 - 20 = 0$ ✓$)$

→ $⑤ \overset{(0)}{\dashrightarrow} ⑦$

$E_{s5} = 20$ $E_{s7} = 21$

$L_{c5} = 20$ $L_{c7} = 20$

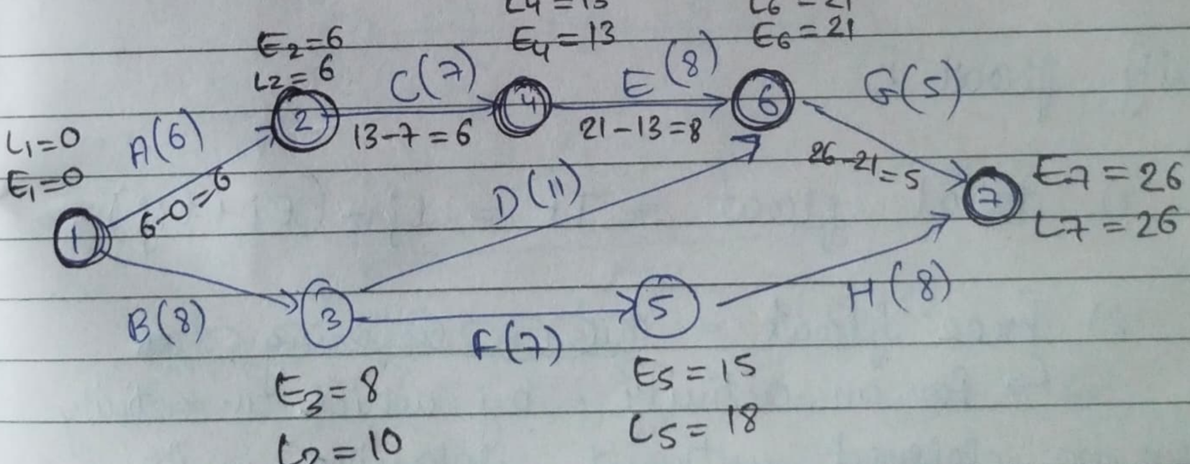
$(21 - 20 \neq 0)$ ✗

∴ Critical Path

$① \xrightarrow{A} ② \xrightarrow{C} ③ \xrightarrow{D} ④ \xrightarrow{E} ⑤ \xrightarrow{\text{Dummy}} ⑥ \xrightarrow{F} ⑦ \xrightarrow{G} ⑧$

Eg ②

A	B	C	D	E	F	G	H
-	-	A	B	C	B	D,E	F
6	8	7	11	8	7	5	8



Critical path $= ① \xrightarrow{A} ② \xrightarrow{C} ④ \xrightarrow{E} ⑥ \xrightarrow{G} ⑦$

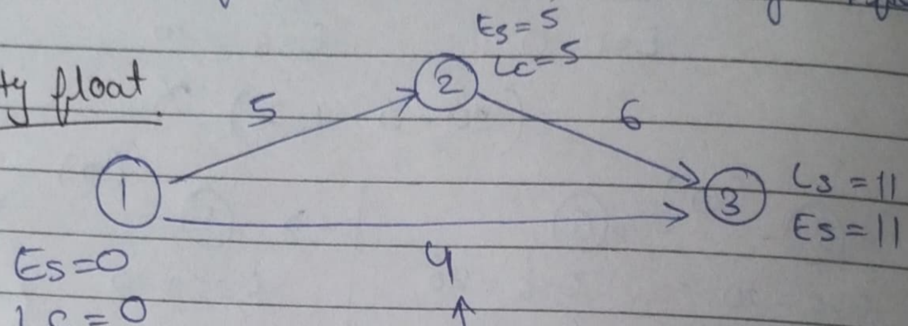
total time $= 26$ (days)

Slack (or Float) of an Activity & Event

↳ The float is the free or delay or gap of time occurence in a project for activity / event.

1) Event float 2) Activity float

<u>activity float</u>



↑

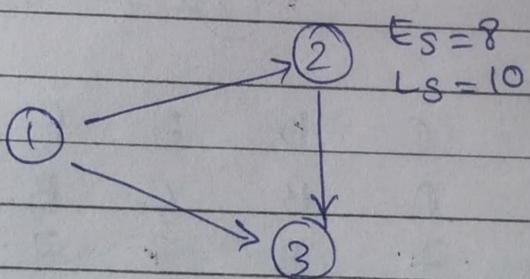
Activity (1-3) there is a gap of time

$E_1 = 0$ $E_3 = 11$

→ (4)

↳ <u>float</u>

<u>event float</u> – $L_i - E_i \geq 0$



$\boxed{L_2 - E_2 = 2}$

Activity float :

1) total float = $TF = L_j - (E_i + t_{ij})$

2) free float = ~~[illegible]~~

↳ for an activity – by which an activity can be delayed without delaying in its immediate successor activities.

$$FF = E_j - (E_i + t_{ij}) \geq 0$$

3) Independent float - It is that portion of total float within which an activity can be delayed for start without affecting floats of the preceding activities.

$$IF = (E_j - L_i) - t_{ij}$$

* IF may be (-ve) then we consider IF as zero.

Remarks

1) $L_i \geq E_i$

$IF \leq FF \leq TF$

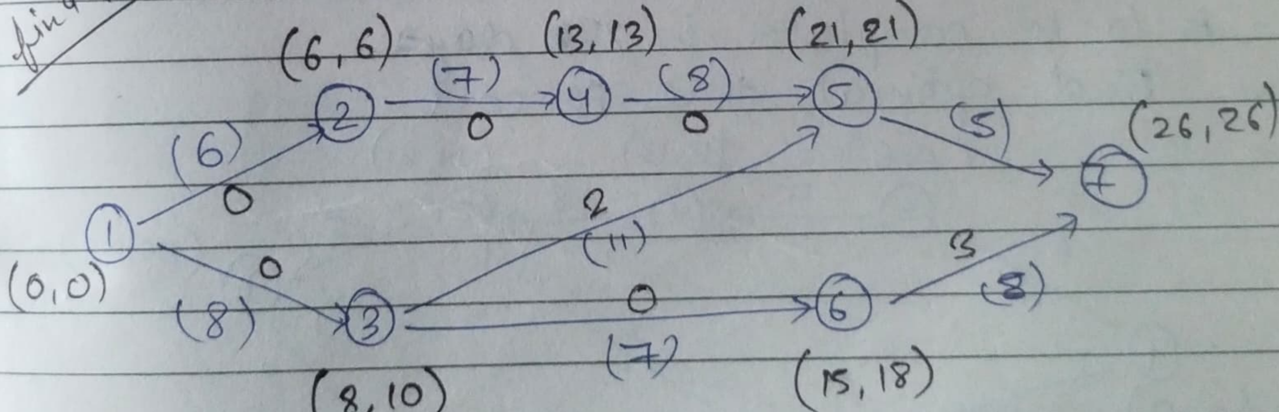
IF → preceding; FF → successor; TF → for all

2) An activity is critical if $TF = 0$

$FF = 0 \not\Rightarrow$ it is a critical activity.

eg. find FF.

1-2	~~2~~-3	2-4	4-5	3-5	3-6	5-7	6-7
6	8	7	8	11	7	5	8



$FF = E_j - (E_i + t_{ij})$

* $3 \xrightarrow{11} 5$ (8,10) (21,21)

→ $21 - (8 + 11)$

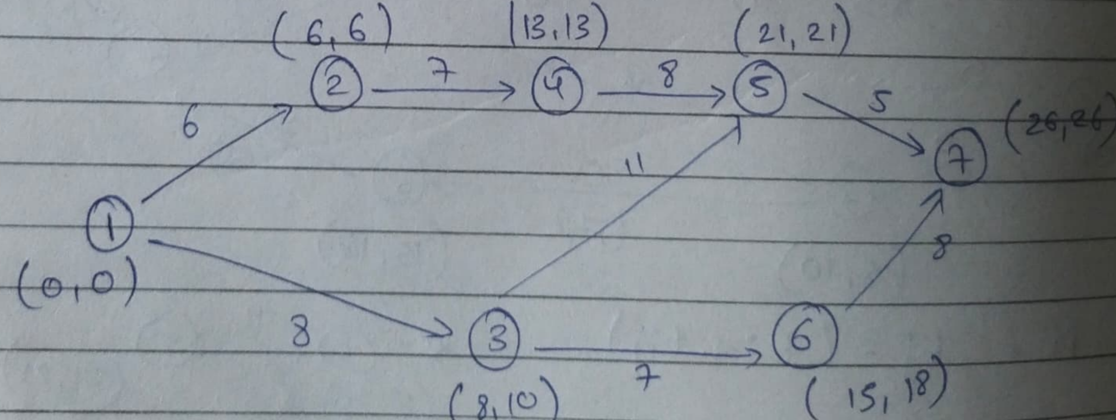
= gap.

Optimizing Project / Resource Scheduling / Crashing in Project Network

Pre-requisite - Network of Project
- free float
- E_S , L_C
- Critical paths

Ques. slope (calculated)	Activities	Normal time	Normal cost	Crashing time	Crash Cost
100	1-2	6	300	5	400
100	1-3	8	400	6	600
100	2-4	7	400	5	600
least $\boxed{87.5}$ in critical path	→ 4-5	8	1000	4	1350
50	3-5	11	500	5	800
100	3-6	7	400	6	500
200	5-7	5	1000	3	1400
150	6-7	8	500	5	950

(1) - Network , ~~tc~~ completion time & cost → ΣNC
(2) - find most economical schedule if project is to be complete in 23 days.
(3) - Find optimal time & cost

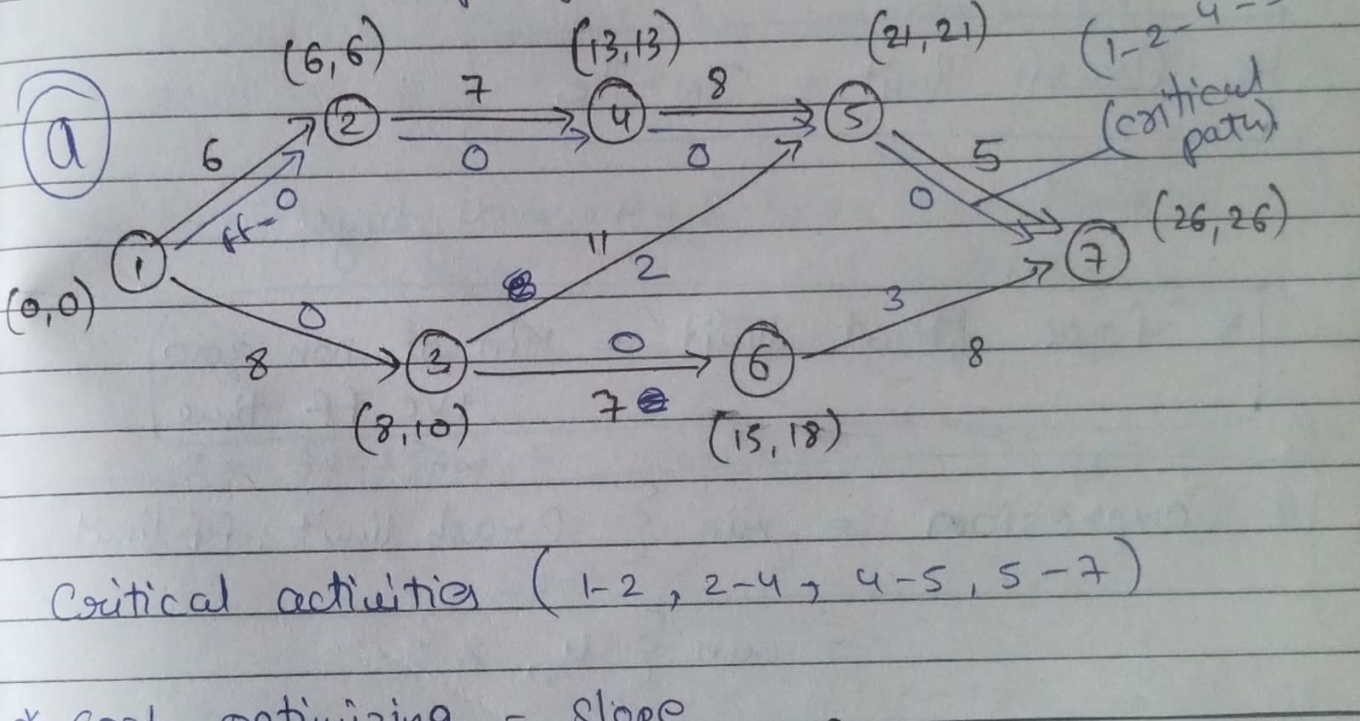


Project completion time = 26 days

Project cost = Σ Normal cost
= 4500 ₹

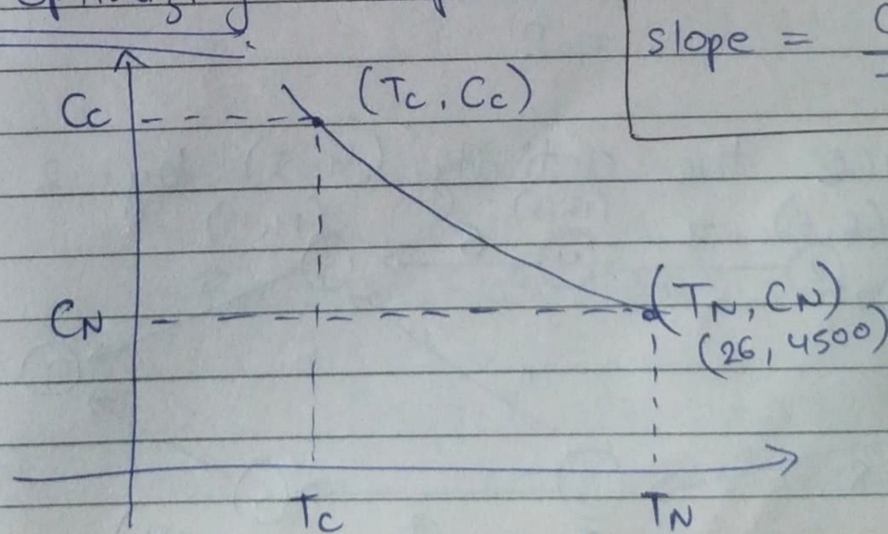
(ii) for Crashing

step 1: Identify critical Path
step 2: activity slope / reduction
step 3: find FF, crash limit, Compression
step 4: Make reduction in selected activities and recalculate project time.



Critical activities (1-2, 2-4, 4-5, 5-7)

* cost optimizing - slope



$$\text{slope} = \frac{C_C - C_N}{T_N - T_C} > 0$$

Slope of (1–2) $= \dfrac{C_c - C_N}{T_N - T_c}$

$= \dfrac{400 - 300}{6 - 5} = \dfrac{100}{1} = 100$

(1–3) $\longrightarrow \dfrac{200}{2} = \dfrac{100}{1} = 100/\text{day}$

(2–4) $\longrightarrow \dfrac{100}{1}$ ~~el~~ so on.

By least slope (4–5) is best for reduction
↳ 87.5

$\boxed{* \text{ Crash limit} = T_N - T_c}$

$= 8 - 4 = 4$ days.

$\boxed{* \text{ free float limit} = \text{Min of non-zero +ve FF time}}$

$\therefore \min\{2, 3\} = 2$

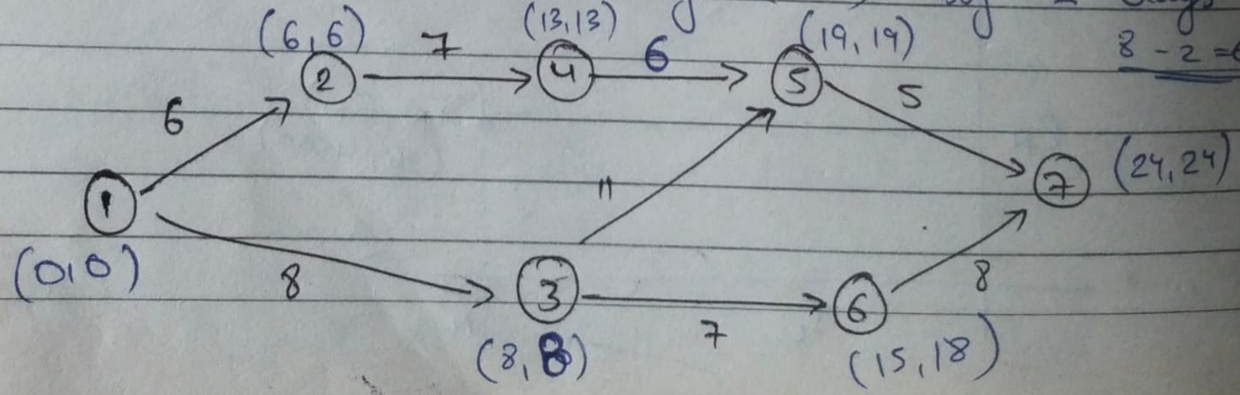
$\boxed{* \text{ Compresion} = \min\{\text{Crash limit, FF limit}\}}$

$= \min\{4, 2\}$

$= 2$

Reduce the activity (4–5) by 2 days.

$8 - 2 = 6$



to reduce more - we have to follow whole process again.

(b) Project time $= 24$ days (reduction)

cost $=$ Normal cost $+ 2$ day

$= 4500 + (87.5 \times 2)$ least slope ↓ activity.

$= 4675$

critical path $1 \to 2 \to 4 \to 5 \to 7$: $(4-5)$

$1 \to 3 \to 5 \to 7$: $(3-5)$

Remark : the common activity in two critical path could be selected for reduction if

slope of common activity $<$ sum of slope of individual activities

$\text{slope}(5-7) \not< \text{slope}(4-5) + \text{slope}(3-5)$

$\therefore \quad 200 \not< 87.5 + 50$

			FF	
crash limit	$(4-5)$	$6-4=2$	1	1
	$(3-5)$	$11-5=6$	1	

compression limit $= \min\{CL, FF\}$

$= 1$

reduce in both path.

$\therefore$ Project time $= 23$ days

cost $= 4675 + (87.5 \times 1) + (50 \times 1)$

$= 4812.5$

(iii) Optimal time & cost.

		Best Slope
Critical paths (5→7)	1 → 2 → 4 → 5 → 7	(4-5)
(1-3)	1 → 3 → 5 → 7	(3-5)
	1 → 3 → 6 → 7	(3-6)

choose (4-5) & (1-3) for reduction

	CL	
(4-5)	5-4 = 1	1 day.
(1-3)	8-6 = 2	

∴ time 22 days

cost = 4812.5 + (87.5) + 100

= 5000

* Repeat process for more optimal, until all activites are reduced once.

Shortest Path Problems:

Algo. - Dijkstra's Algo.

- A* Algo.

Network flow Problems

a) Max. flow Problems - find max. possible flow from a source node to a sink node in a flow network.

- ford fulkerson method
- Edmonds karp Algo.

b) Minimum Cut Problem - identify minimum set of edges that, if ~~set~~ removed, would disconnect the source from the sink.
- Ford fulkerson method

Ford fulkerson Algo. for Maximum flow problem :

Problem : Given graph which represents a flow network where every edge has a capacity. Also given two vertices source (S) and sink (t) in the graph. Find out the maximum possible flow from s to t with following constraints :

a) Flow of edge does not exceed the given capacity of edge.
b) In-flow is equal to out-flow for every vertex except s and t.

Algo. Ford fulkerson
1) Start with a initial flow as 0
2) While there is an augmenting path from s to t, add two path flow to flow
3) Return flow.

Terminologies

① Residual Graph - It's a graph which indicates additional possible flow. If there is such path from s to t then there is a possibility to add flow.

② Residual Capacity - It's original capacity of the edge minus flow.

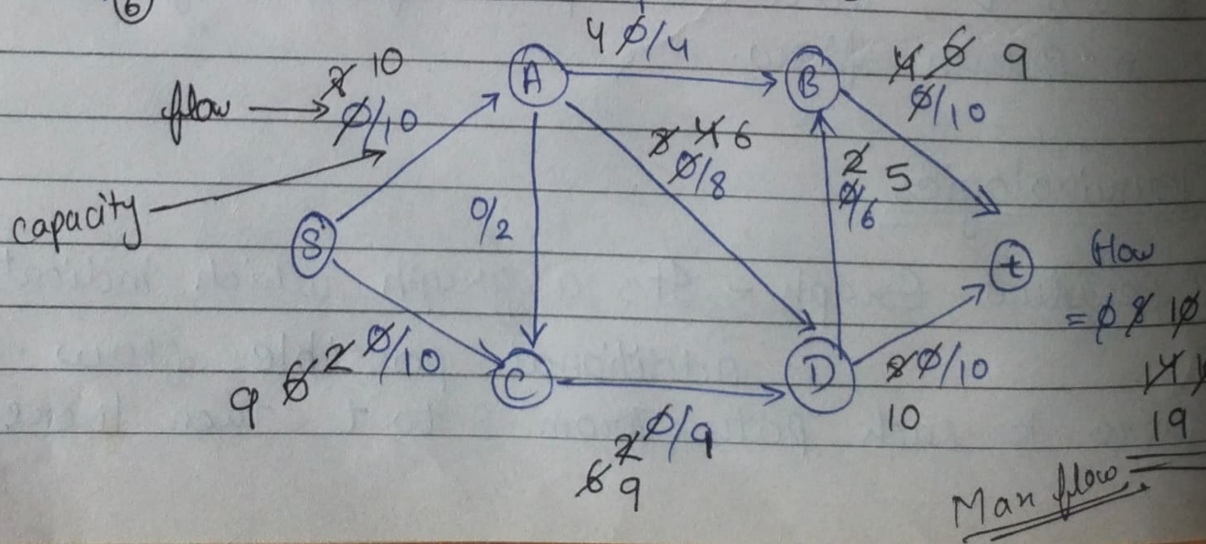
③ Minimul cut - Also known as Bottle neck capacity, which decides maximum possible flow from s to t through an augmented path.

④ Augmenting path - Augmenting path can be done in two ways

a) Non-full forward edges
b) Non-empty backward edges

Max-flow example (Ford Fulkerson)

Computing flow.	Augmenting path	Bottle Neck capacity
0	Residual	
8	① $S \xrightarrow{2} A \xrightarrow{0} D \xrightarrow{2} T$	8
10	② $S \xrightarrow{8} C \xrightarrow{7} D \xrightarrow{0} T$	2
14	③ $S \xrightarrow{4} C \xrightarrow{2} D \xrightarrow{4} A \xrightarrow{0} B \xrightarrow{6} T$	4
16	④ $S \xrightarrow{0} A \xrightarrow{2} D \xrightarrow{4} B \xrightarrow{4} T$	2
19	⑤ $S \xrightarrow{1} C \xrightarrow{0} D \xrightarrow{1} B \xrightarrow{1} T$	3
	⑥	



complexity → $O(E \times F)$

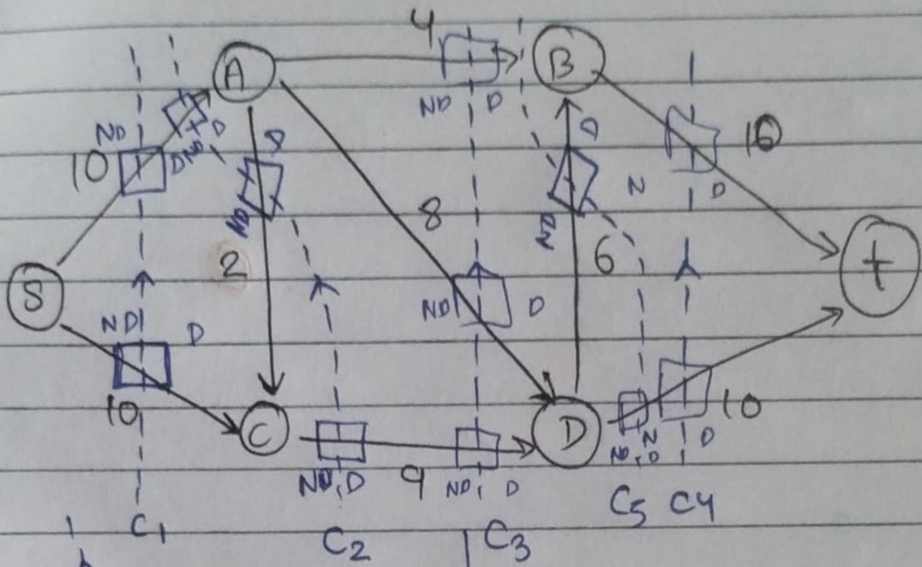
E = no. of edges

F → Max. flow

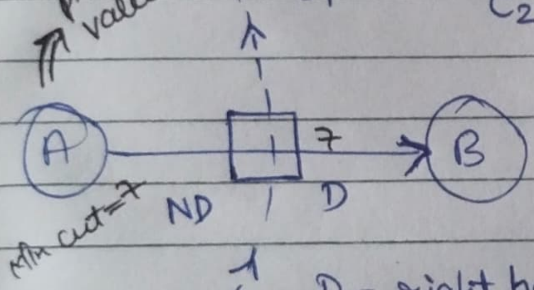
Min cut problem

Min Cut = Max flow

→



pick value



D - right hand

ND - left hand

Direction of car

↳ bottom to top

ND → ∞

D → 1

① If Arrow towards ND

↳ we will pick the value

② If Arrow is towards D

↳ we will not pick the value.

$C_1 = 10 + 10 = 20$

$\boxed{C_2 = 9 + 0 + 10 = 19}$

$C_3 = 9 + 8 + 4 = 21$

$C_4 = 10 + 10 = 20$

$C_5 = 10 + 6 + 4 = 20$

Min cut = 19 / Ans.