Pintu Sir

Lecture-1

TOPIC NAME : Algorithm Analysis & Design

DAY : Tuesday

TIME :

DATE : 8 / 8 / 23

Objective :

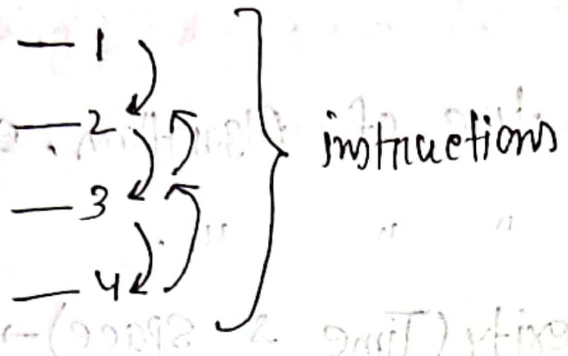
1. Study different type of Algorithm,
2. Analyze " " " ".
3. Analyze → complexity (Time & Space) → Quantitive analyze
   → Document / clear
4. Devise a new Algorithm.
   → check complexity (Time & Space)
   → Document (check Ambiguous)

Problem
→ Unstructured
→ Structured → Analytical Model

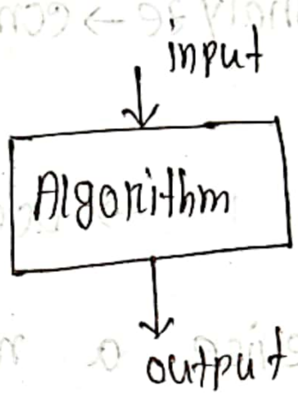
Algorithm : An algorithm is a finite set of instructions that is followed, accomplishes a particular task.

particular → specific

TOPIC NAME : DAY: TIME: DATE: / /



Properties :



1. Input
→ Zere on more quantities externaly supplied.

2. Output
→ at least one quantity is produced.

3. Definiteness
→ Each instruction is clear and unambigious.

4. Finiteness
→ Must terminate

5. Effectiveness
→ each instruction must have a principle.

Pintu Sir

Lecture-2

TOPIC NAME : Introduction

DAY : Wednesday

TIME :

DATE : 16/ 8 /23

Real life algorithm example :

1. Traffic light.
2. Sensor in office.
3. Microcontroller devices (washing machine, light on/off device)

→ intelligent Algorithm

→ one stop service

Polimi.it

Lecture-3

Pintu Sir

Introduction

Wednesday

23/8/23

Real life example of algorithm :

4. Sorting documents and papers.

→ Bucket sort

5. Searching Book in the library

TOPIC NAME : DAY : TIME : DATE : / /

Steps of algorithms:

1. How to devise algorithm.
   innovate

2. How to validate algorithm.
   → যেই Prob solve এর জন্য algo develop করেছি, তা solve করছে কি না।

   → verification : একটু change করে, আবার check করা।

3. How to analysis algorithm.

4. How to test a program.

5. Testing
   → Performance analysis (time, space কত লাগছে)
   → " measurement
   → Post

Space complexity

1. Fixed Part
2. Variable Part

TOPIC NAME : DAY : TIME : DATE : / /

Fixed part :

① Instruction space
② Space for simple variables
③ Fixed size
④ Fixed size for constant

Variable part :

→ input এর সাথে related

① Reference variable ② Recursion step

n data sum.

↓

1 word

$S(n) \geq n+3$ word

i

n

sum

Time complexity

① compile time ⟶ independent ⟶ ignored

\+

② Execution time ⟶ Dependent

→ count Program steps ⟶ Arithmetic operations
⟶ loop (n+1)
⟶ Decision making

$\log n$, n^2

Pintu Sir

TOPIC NAME : Algorithm Design Paradigm/ Strategy

Lecture-4

DAY : Tuesday

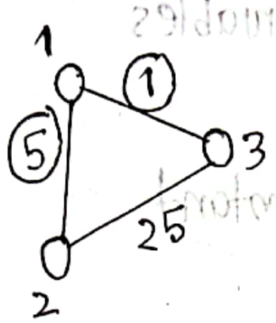
TIME :

DATE : 12/ 9 / 23

Methods:

1. Greedy Method:

⇒ current situation



1–2

1 +25 = 26

(i) Fractional Knapsack

(ii) Activity selection Problem

(iii) Minimum cost Spanning Tree:

2. Divide and conquere:

⇒ Works Recursively

3. Dynamic Programming:

⇒ Problems overall deseision

DP ⟷ Greedy

(i) 0/1 Knapsack

(ii) Sum of subset problem

4. Linear programming:

⇒ AKA Optimization Algorithm

(i) Traveling salesman Problem.

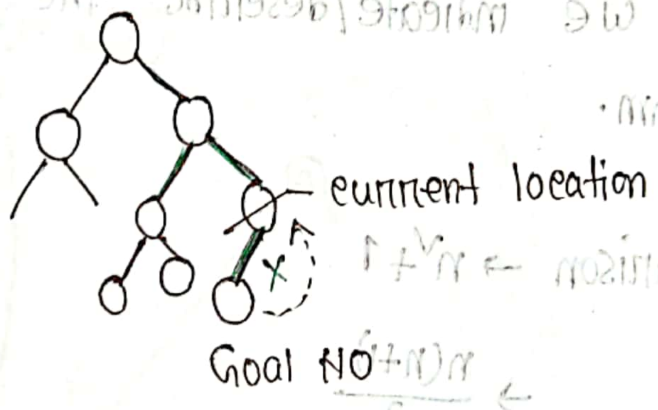
5. Reduction :

⇒ AKA Tramform and conqure.

⇒ Unknown → known → conqure → merge

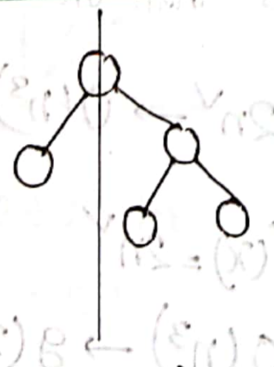
6. Back Tracking:

State Space Tree



N -
(i) ~~Ant~~ Queen problem (ii) Maize Problem

7. Branch and Bound :



Max → 100
Min → 20 → Max, Min এর বাহিরে গেলে ওই Branch এ যাব না।

(i) Job sequencing Problem
(ii) Traveling Salerman Problem (TSP)

TOPIC NAME :

DAY :

TIME : DATE : / /

8. Implemental :

9. Randomist / Probabilistic :

Asymptotic Notations

How we indicate/describe the running time of algorithm.

comparison → n^2+1

→ $\frac{n(n+1)}{2}$

→ $2n-1$

① O notation : Asymptotic "less than or equal"

$f(x) \le g(x)$ # $2n^2 \to O(n^3)$

$O(\underset{g(x)}{\downarrow}) \Rightarrow O(n^2)$

$\downarrow$

$O(n^3)$

$f(x) = 2n^2$

$O(n^3) \to g(x)$

② Ω notation: Asymptotic "greater or equal"

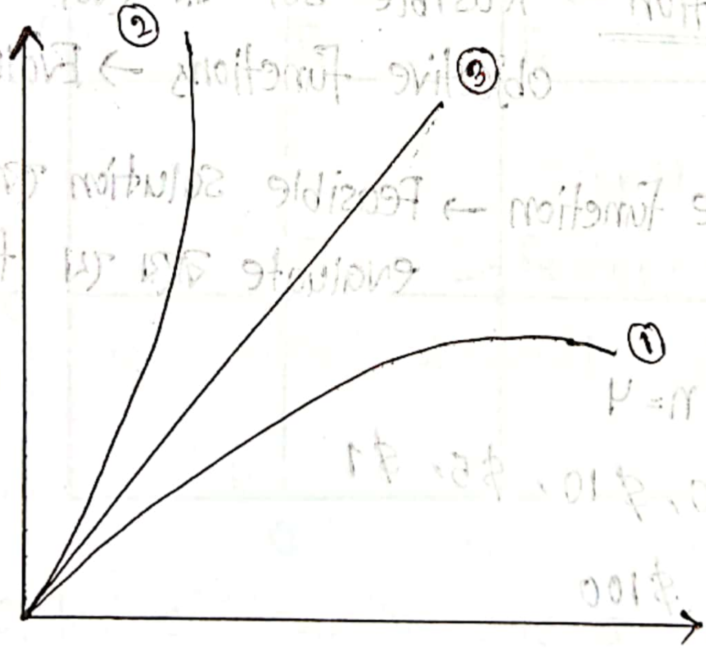
$$F(x) \geq g(x)$$

$$n^3 \rightarrow \Omega(n^2)$$

$$n \text{ vs } \log n \rightarrow \Omega(\log n)$$

③ Θ notation: Asymptotic "Equal"

$$F(x) = g(x)$$



$1+2+3+4+ \text{----} n$

$$= \frac{n(n+1)}{2} = \frac{n^2}{2} + \frac{n}{2} \quad \therefore \Theta(n^2)$$

Pintu Sir

Lecture-5

TOPIC NAME : Methods

DAY : Wednesday

TIME :

DATE : 13 / 9 / 23

Greedy Methods

Problem configuration → n inputs

different choise
collections
values to find

Goal → Obtain a subset that satisfy some constrains

Definition

Feasible solution → constrains satisfy

Optimal solution → feasible soln এর মধ্যে ভালো ।
objective function's → Evaluate → best

Objective function → Feasible solution কে evaluate করে যে function.

Example : $n = 4$

\$30, \$20, \$10, \$5, \$1

constrains → \$100

Fractional Knapsack Problem :

n=4 | p_i / b_i → A positive benefit / profit

w_i → weight

* item = $p_i/b_i + w_i$

Goal → item → Maximize → Profit

constrains → Total weight

* 0,1 knapsack problem → Dynamic program.

$$\text{Objective} = \sum_{i=1}^{n} p_i x_i$$

↓

1, ------ 1/2

$$\text{Subject to} = \sum_{1 \le i \le n} w_i x_i \le m$$

$$X = (x_1, x_2, x_3 \text{ ------ } x_n)$$

$x_1 \to 1$, $x_2 \to 1$, $x_3 \to 1/2$, ------ $\to 1$, $x_n \to 0$

Solution vector –

① Fixed Toupple

② Variable Toupple

TOPIC NAME : DAY : TIME : DATE : / /

$n=3$, $m=20$

$\downarrow$ $\hookrightarrow$ Total weight

No of item

$p=(p_1, p_2, p_3)=(25, 24, 15)$

$w=(w_1, w_2, w_3)=(18, 15, 10)$

Feassible Solution:

	$\sum w_i x_i$	$\sum p_i x_i$
1. $(1/2, 1/3, 1/4)$	16.5	24.25
2. $(1, 2/15, 0)$	20	28.5
3. $(0, 2/3, 1)$	20	31
4. $(0, 1, 1/2)$	20	31.5 $\rightarrow$ optimal

$w=(4,1,2,6,1)$ $m=10$

$p=(12, 32, 40, 30, 50)$ $n=4$

Pintu Sir

Lecture-6

TOPIC NAME : Minimum Cost Spanning Tree

DAY : Tuesday

TIME :

DATE : 3 / 10 / 23

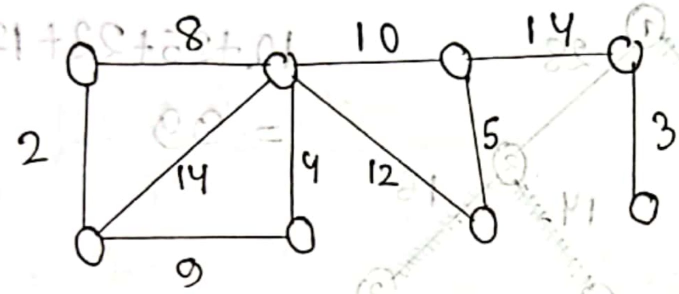
let $G = (V, E)$

$E \to$ Edges

$V \to$ vertex

Subgraph / Tree = $t = (V ; E')$

#



$e = n - 1$

$\Rightarrow$ 8 vertex

$e = n - 1 = 8 - 1 = 7$

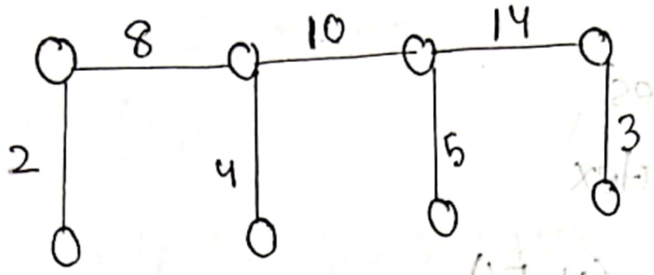
Edges = $10 - 3 = 7$

Algorithm :

Start with a 1-vertex tree and grow it into a n-vertex tree by

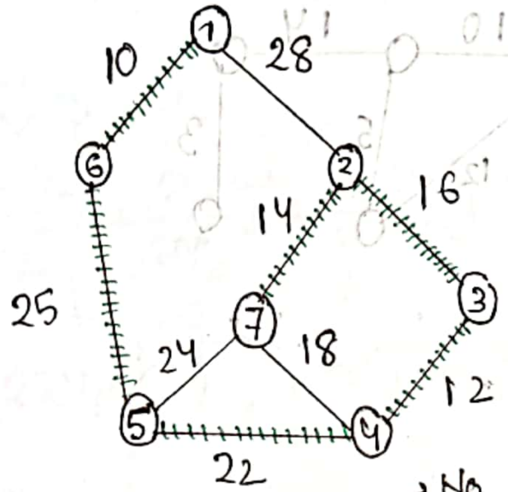
→ Repeatedly adding a vertex and an edge

→ When there is a choice, add a best cost edge



$2+8+4+10+5+14+3$
$= 48$

#



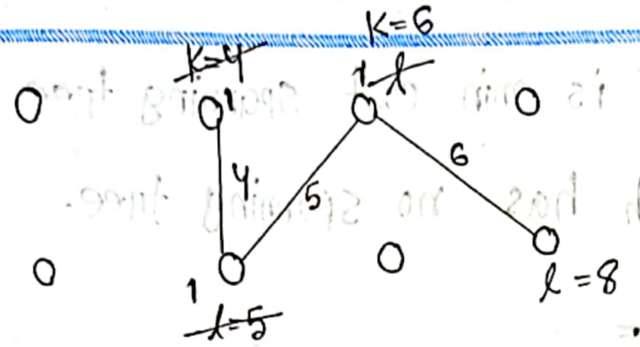
$10+25+22+12+16+14$
$= 99$

Algorithm: Prim (E, cost, n, t)

- n → No of edges
- t → Minimum Spanning Tree
- cost → cost matrix [1 : n-1, 1]
- E → Set of edges in graph
- Prim → Book: Shani

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & & & & & \\ 2 & & & & & \\ 3 & & & & & \\ 4 & & & & & \\ \vdots & & & & & \end{array}$$

4	5
6	7

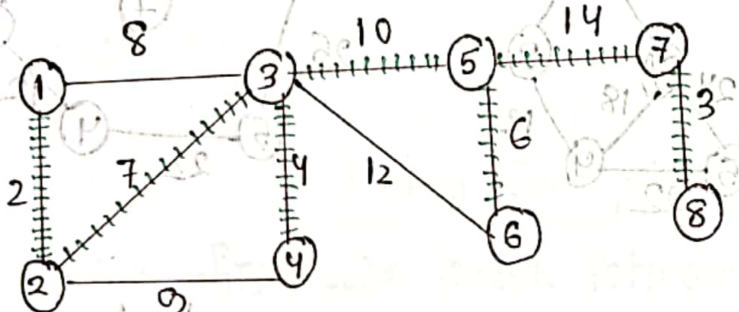


Total cost = 4

Lecture- 7
wednesday
4/10/23

Pintu Sir

Kruskal's Algorithm



2+7+4+10+6+14+3
=46

Pseudo code :

Start with an empty set T of edges (T → Tree edge)

while (E is not empty && |T| != n-1) (E ↳ Graph edges)

{

let (u,v) be a least cost Edge in E

E= E - {(u,v)}

if (u,v) does not create a cycle in T

Add edge (u,v) to T

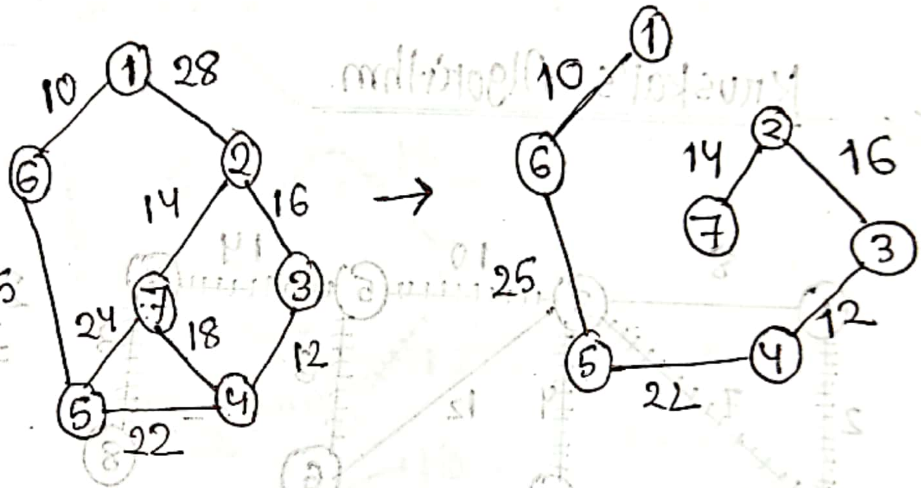
}

if ($|T| == n-1$) ; T is min cost spanning tree

else Graph has no spanning tree.

Print mincost =

#



Lecture-8
Tuesday
10/10/23

Pintu Sir

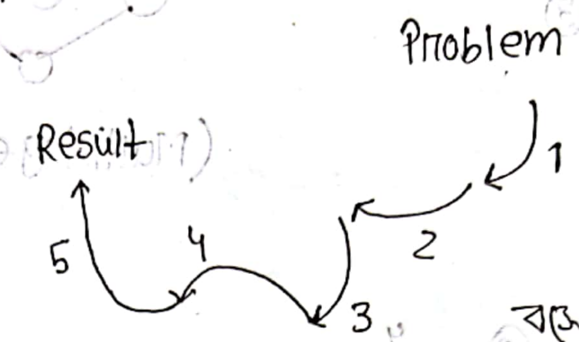
Divide and Conqure Strategy

Binary Search
Quick Sort
Mergesort

Algorithm + Complexity Analysis

Dynamic Programming

→ Strategy / Paradigm



* sequence maintain করা হয়।

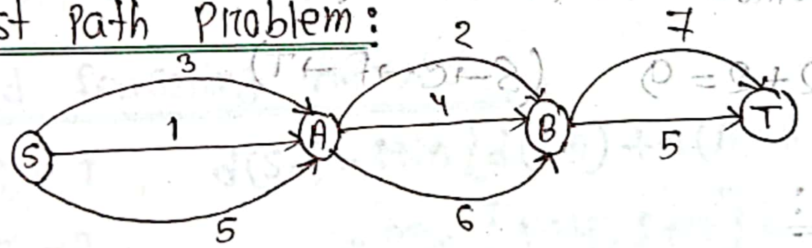
* sequence 1 এ বসে sequence 5 এ কি হবে তা think করতে পারবে।

0/1 Knapsack Problem:

x_1	x_2	x_3	x_4
1	0	0	1

$x_i = 0/1$

Shortest Path Problem:



[multistage graph]

Greedy method = $1 + 2 + 5 = 8$

* Graph ⟶ Multistage Graph

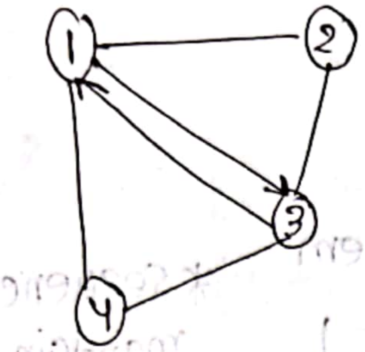
Multistage Graph: $G = (V, E)$ → directed graph

└→ Partitioned into $k \geq 2$ sets

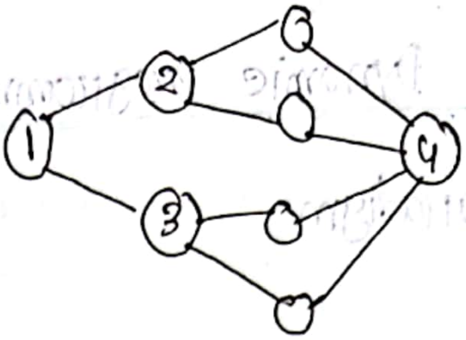
$\Rightarrow V_i \quad 1 \leq i \leq k$

$\Rightarrow |V_1| = |V_k| = 1$

#

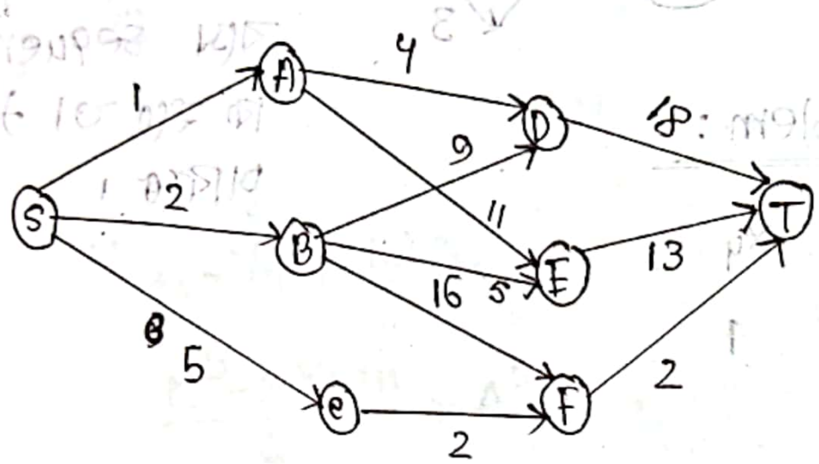


(Graph)



(Multistage Graph)

#



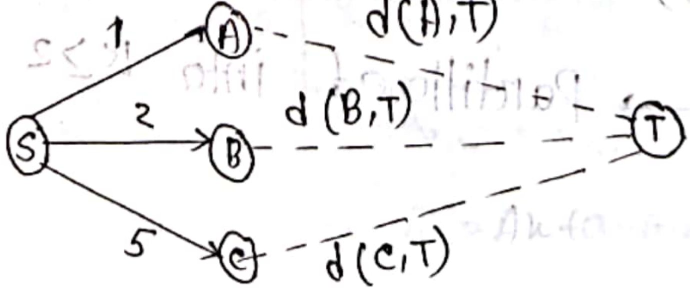
Greedy method = $1+4+18 = 23$ $(S \to A \to D \to T)$

DP = $5+2+2 = 9$ $(S \to C \to F \to T)$

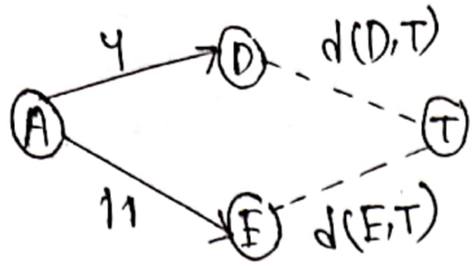
Two methods :

1. Backward Reasoning Approach
2. Forward " "

1. Backward Reasoning Approach :



$$d(S,T) = \text{Min}\{1+d(A,T),\ 2+d(B,T),\ 5+d(C,T)\}$$

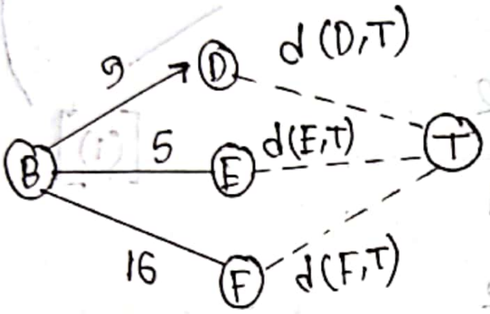


$d(A,T) = \text{Min}\{4 + d(D,T); 11 + d(E,T)\}$

$= \text{Min}\{4+18,\ 11+13\}$

$= 22$

$d(S,T) = \text{Min}\{1+22,\ 2+18,\ 5+4\} = 9$ → Result



$d(B,T) = 18$

$d(C,T) = 4$

2. Forward Reasoning Approach:

$d(S,A) = 1$

$d(S,B) = 2$

$d(S,C) = 5$

$d(S,D) = \text{Min}\{d(S,A) + d(A,D),\ d(S,B) + d(B,D)\}$

$= \text{Min}\{1+4,\ 2+9\} = 5$

$d(S,E) = \text{Min}\{d(S,A) + d(A,E),\ d(S,B) + d(B,E)\}$

$= \text{Min}\{1+9,\ 2+5\} = 7$

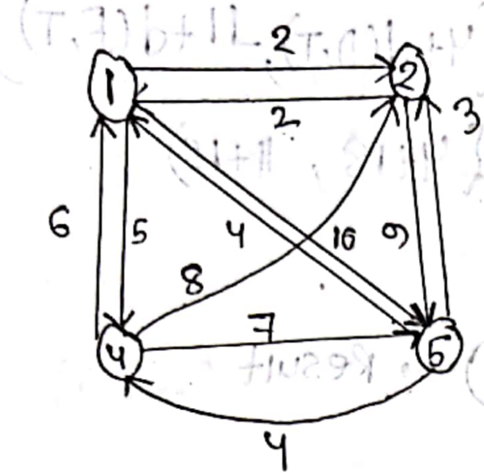
$d(S,F) = \text{Min}\{d(S,B) + d(B,F),\ d(S,C) + d(C,F)\}$

$= \text{Min}\{2+16,\ 5+2\} = 7$

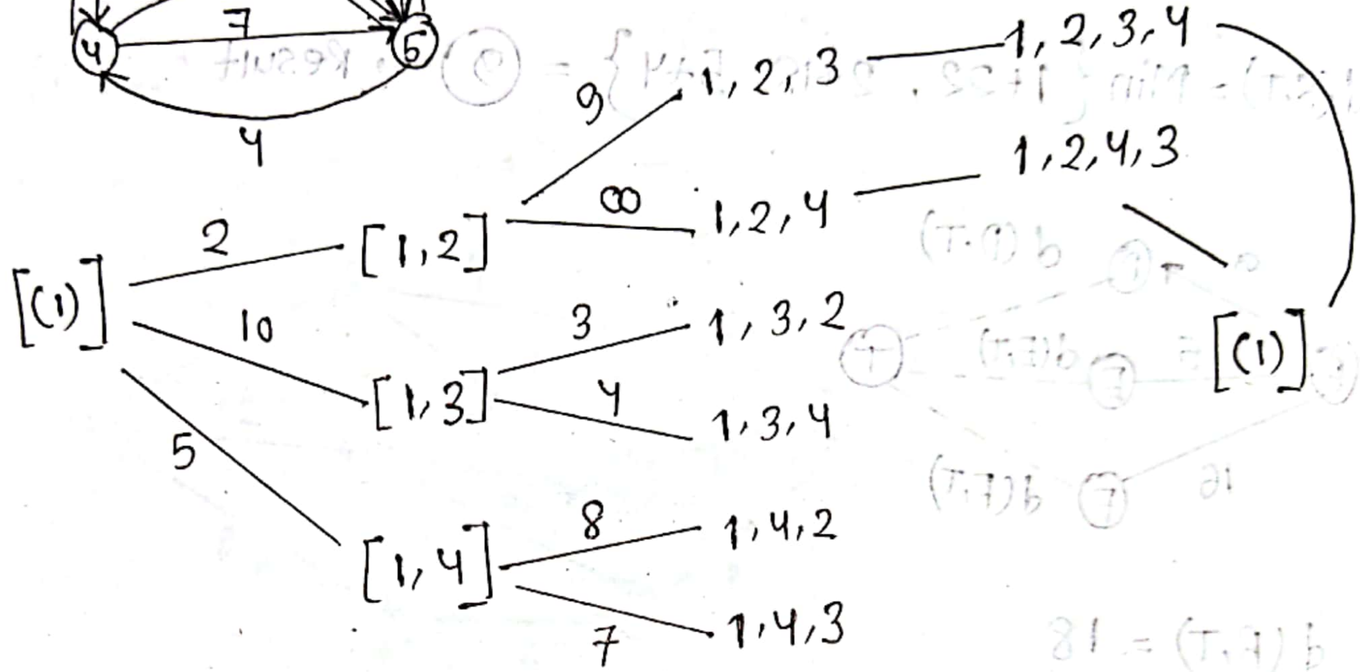
$d(S,T) = \text{Min}\{$

$= 9$

#



1
2
4
5
2
2
3
6
5
4
16
9
8
7
4



[(1)]
2
10
5
[1,2]
[1,3]
[1,4]
9
∞
3
4
8
7
1,2,3
1,2,4
1,3,2
1,3,4
1,4,2
1,4,3
1,2,3,4
1,2,4,3
[(1)]