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

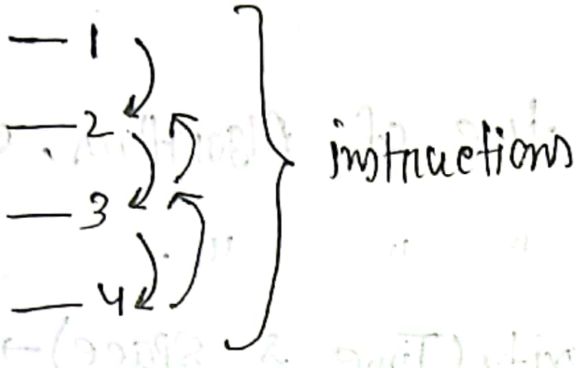
↓ Unstructured

↓ Structured → Analytical Model

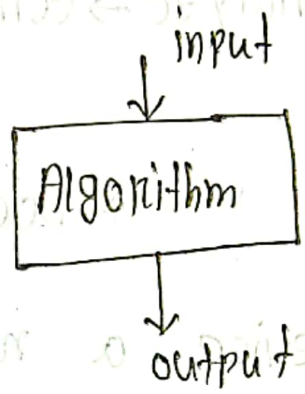
Algorithm : An algorithm is a finite set of instructions that is followed, accomplishes a particular task.

particular ↓ specific

TOPIC NAME : DAY : TIME : DATE : / /



Properties :



1. Input
   → Zero or 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.

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

Wednesday

23/8/23

Pintu Sir

Introduction

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.
   → এই 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 parat :

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

Variable part :

→ input এর সাথে related

① Referenee 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)
⟶ Descision making

logn, n^2

Pintu Sir

Lecture

TOPIC NAME : Algorithm Design Paradigm/ Strategy

DAY : Tuesday

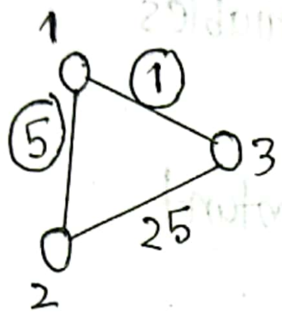
TIME :

DATE : 12/9

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 conqure:

⇒ 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.

TOPIC NAME : DAY : TIME : DATE : / /

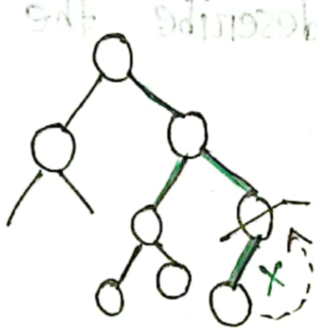
5. Reduction :

=> AKA Tramform and conqure.

=> Unknown → known → conqure → merge

6. Back Tracking:

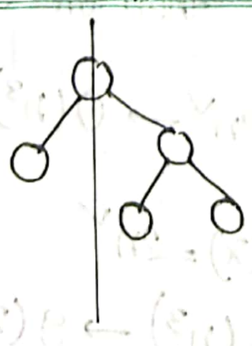
State Space Tree



N -

(i) ~~Ant~~ Queen Problem (ii) Maize Problem

7. Branch and Bound :



Max → 100

Min → 20 → Max, Min এর বাইরে গেলে ঐ Branch এ যাব না।

(i) Job sequencing Problem

(ii) Traveling Salesman Problem (TSP)

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"

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

$O(\ \downarrow\ )$ ⇒ $O(n^2)$ # $f(x) = 2n^2$

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

TOPIC NAME : DAY : TIME : DATE : / /

② Ω notation: Asymptotic "greater or equal"

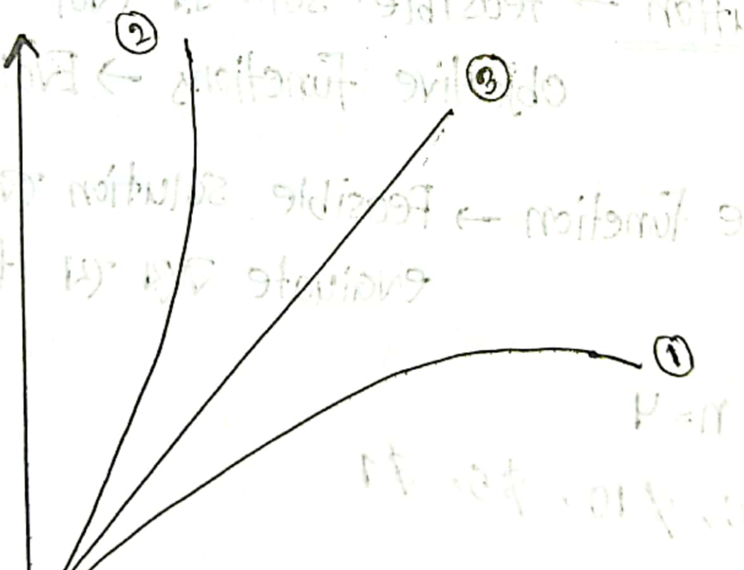
$$f(x) \geq g(x)$$

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

n vs $\log n \rightarrow \Omega(\log n)$

③ Θ notation: Asymptotic "Equal"

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



$1+2+3+4+ \cdots n$

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

Pintu Sir

TOPIC NAME : Methods

Lecture-5

DAY : Wednesday

TIME :

DATE : 13 / 9

Greedy Methods

Problem configuration → n inputs

different choise
collections
values to find

Goal → Obtain a subset that satisfy some constrains

Definition

- Feasible solution → constrain satisfy
- Optimal solution → feasible soln এর যেটা ভালো।
  objective function's → Evaluate → best

Objective Function → Feasible solution কে evaluate করে যে function.

Example: n = 4

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

constrain → \$100

TOPIC NAME : DAY : TIME : DATE : / /

Fractional Knapsack Problem :

$n=4$

$p_i / b_i \longrightarrow$ A positive benefit/ profit

$w_i \longrightarrow$ weight

$*$ item = $p_i/b_i + w_i$

Goal → item → Maximize → Profit

contrains → Total weight

$*$ 0,1 knapsack problem ⟶ Dynamic program.

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

x_i ↓ 1, ------ ½

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

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

$x_1 \to 1$, $x_2 \to 1$, $x_3 \to ½$, ... $\to 1$, $x_n \to 0$

Solution vector -

① Fixed Toupple

② Variable Toupple

TOPIC NAME : DAY : TIME : DATE : /

$n=3$, $m=20$ ↳ 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 → optimal

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

$P=($~~[illegible]~~ $12, 32, 40, 30, 50)$ $n=4$

Lecture-6

TOPIC NAME : Minimum Cost Spanning Tree

DAY: Tuesday

TIME:

DATE: 3 /10 /23

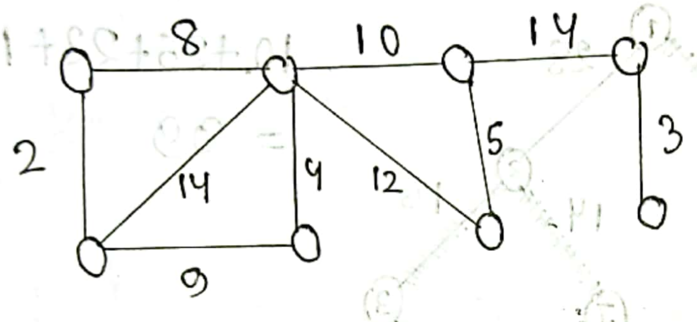
let $G = (V, E)$

E → Edges

V → vertex

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

#



$e = n - 1$

⇒ 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

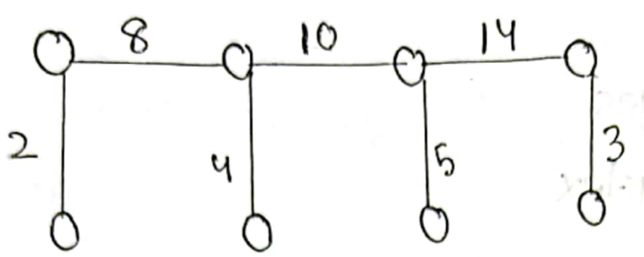
→ Repeatedly adding a vertex and an edge

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

TOPIC NAME :

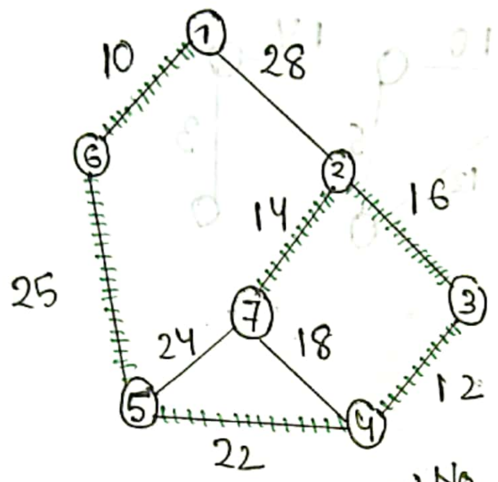
DAY :

TIME : DATE : /



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

$= 48$



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

$= 99$

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

- n → No of edges
- t → Minimum Spanning Tree
- cost → cost matrix
- E → Set of edges in graph

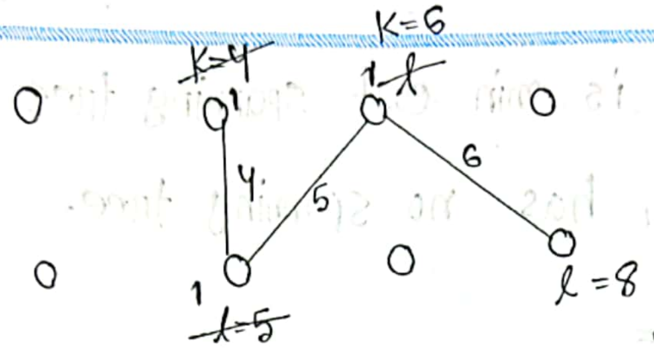
Book: Shani

$[1 : n-1, 1]$

4	5
6	7

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

TOPIC NAME :
DAY :
TIME : DATE : / /

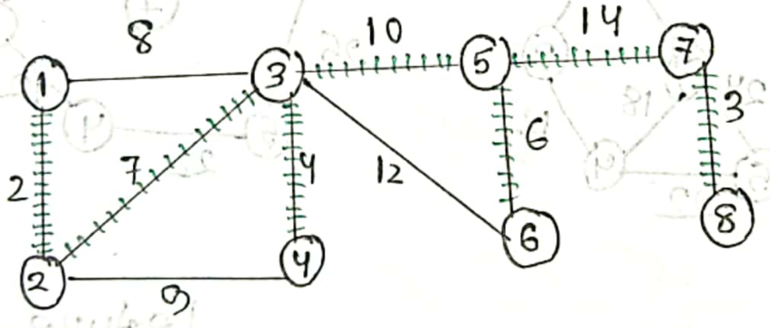


Total cost = 4

Lecture- 7
Wednesday
4/10/23

Pintu Sir

Kruskal's Algorithm



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

Pseudo code :

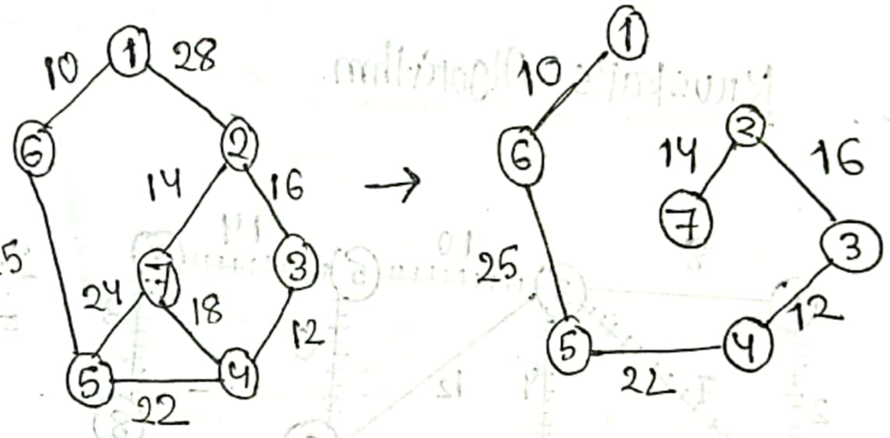
Start with an empty set T of edges → Tree edge
while (E is not empty && |T| != n-1) ↳ Graph edges
{
let (u,v) be a least cost Edge in E
E = E - {(u,v)}
if (u,v) does not create a cycle in T
Add edge (u,v) to T
}

TOPIC NAME : DAY : TIME : DATE : / /

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

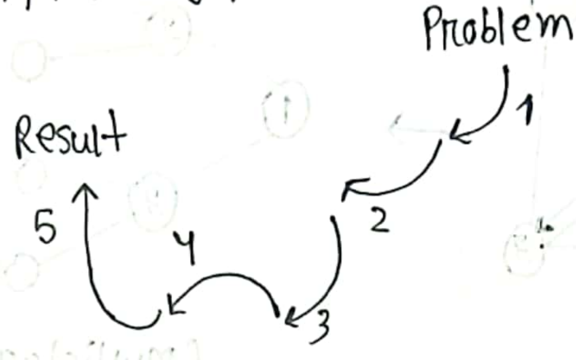
Algorithm + Complexity Analysis

GOOD LUCK™

Lecture-8

TOPIC NAME: Dynamic Programming
DAY: Tuesday
TIME:
DATE: 10/10/23

→ Strategy / Paradigm



* Sequence maintain করা হয়।

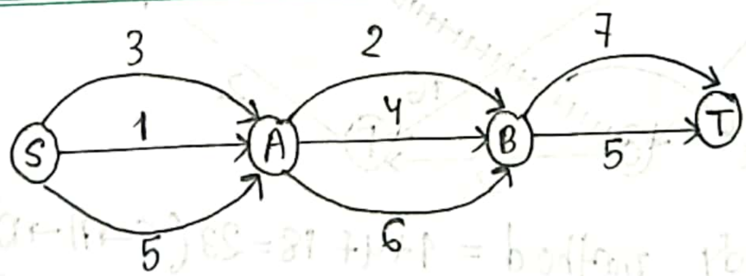
* Sequence 1 এ বসে sequence 5 এ কী হবে তা assume করতে পারবে।

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.

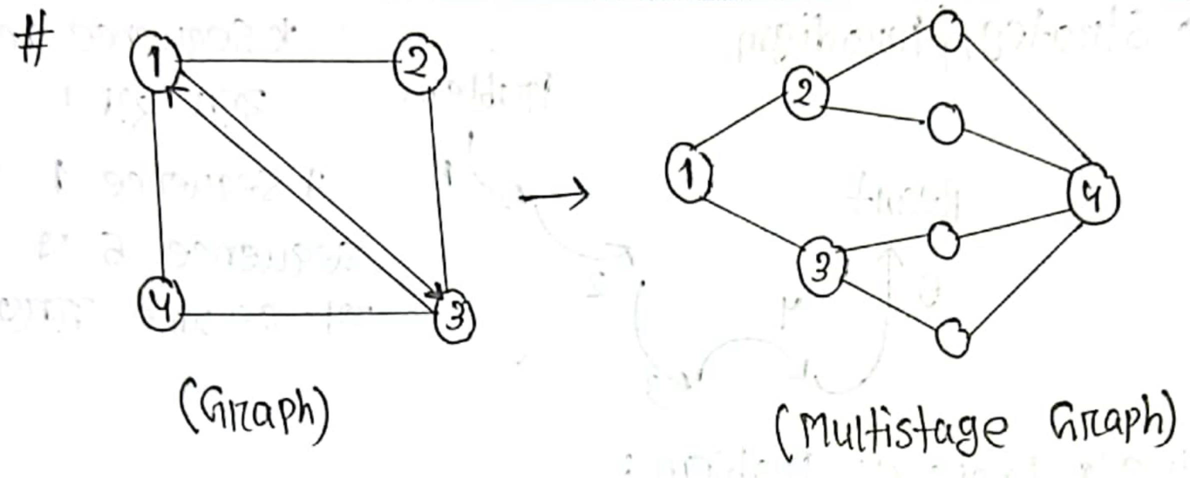
Multistage Graph: $G = (V, E)$ → directed graph

↳ Partitioned into $K \geq 2$ sets

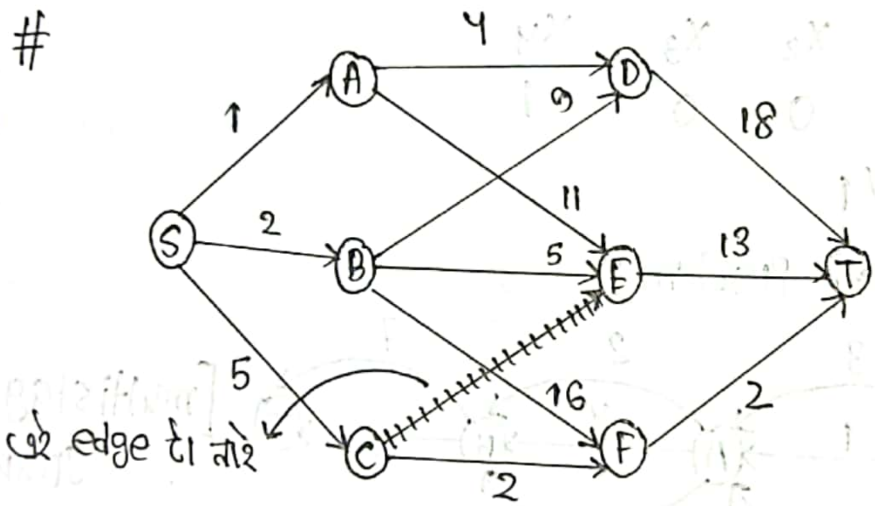
⇒ $V_i \quad 1 \leq i \leq K$

⇒ $|V_1| = |V_K| = 1$

#



#



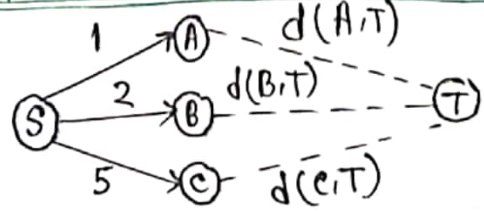
$\Rightarrow$ Greedy method $= 1+4+18 = 23 \; (S \to A \to D \to T)$

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

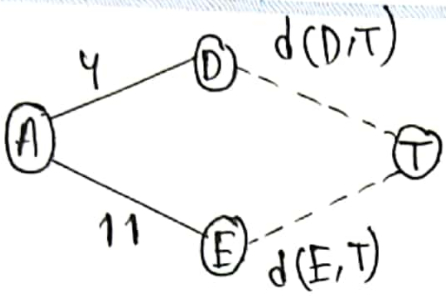
Two methods:

1. Backward Reasoning Approach
2. Forward "   "

1. Backward Reasoning Approach:



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

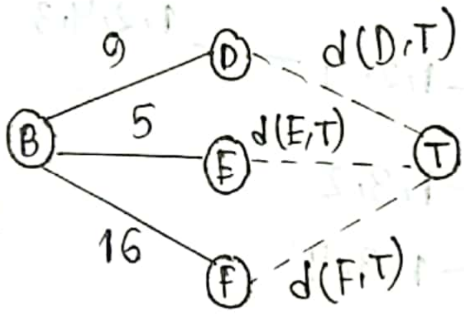


$d(A,T) = Min\{4+d(D,T), 11+d(E,T)\}$

$= Min(4+18, 11+13)$

$= 22$

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



$d(B,T) = 18$

$d(C,T) = 4$

2. Forward Reasoning Approach :

$d(S,A) = 1$

$d(S,B) = 2$

$d(S,C) = 5$

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

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

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

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

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

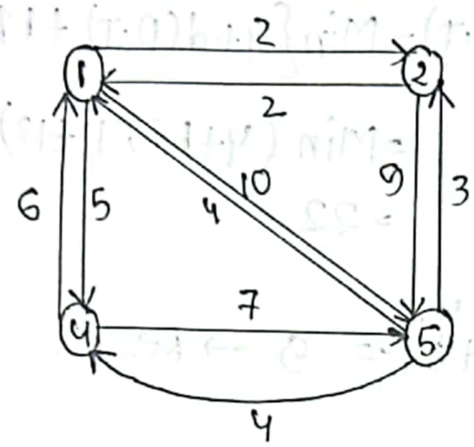
$= Min\{2+16, 5+2\} = 7$

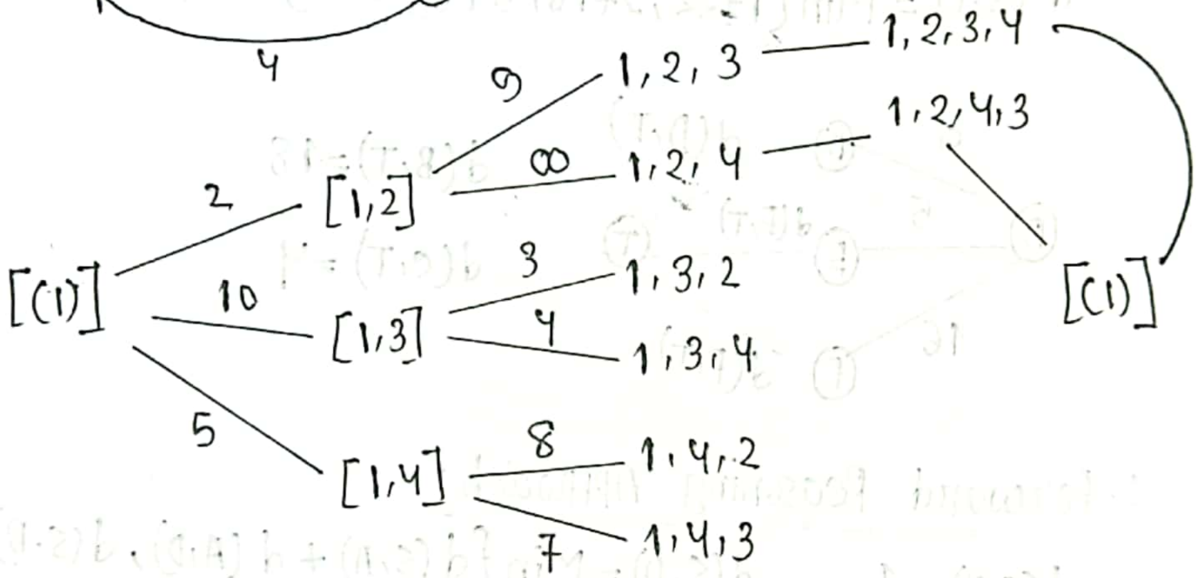
$d(S,T) = Min\{18+d(S,D), 13+d(S,E), 2+d(S,F)\}$

$= 9$

TOPIC NAME : DAY : TIME : DATE : / /

#



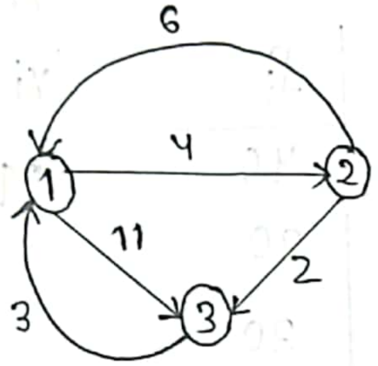


Lecture- 9

TOPIC NAME : Dynamic Programming DAY : Tuesday

TIME : DATE : 17 / 10 / 23

All pair shortest Path Program:



1-2 →
2-1 →
3-1 →
1-3 →

⇒ input = Directed graph

⇒ output = matrix

Adjacency matrix:

K = no of nodes A^K

$$A^0 = A[i,J] = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 4 & 11 \\ 2 & 6 & 0 & 2 \\ 3 & 3 & \infty & 0 \end{array}$$

$$A^1 = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 4 & 11 \\ 2 & 6 & 0 & 2 \\ 3 & 3 & 7 & 0 \end{array}$$

K=1 ------- K = Number of node

for (i ------- i ≤ no of node

for (j = 1 ; j <= no of node

$$A[i,J] = \text{Min}\{A[i,J],\ A[i,K] + A[K,J]\}$$

$$A^2 = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 4 & 6 \\ 2 & 6 & 0 & 2 \\ 3 & 3 & 7 & 0 \end{array}$$

$$A^3 = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 4 & 6 \\ 2 & 6 & 0 & 2 \\ 3 & 3 & 5 & 0 \end{array}$$

TOPIC NAME: DAY: TIME: DATE: / /

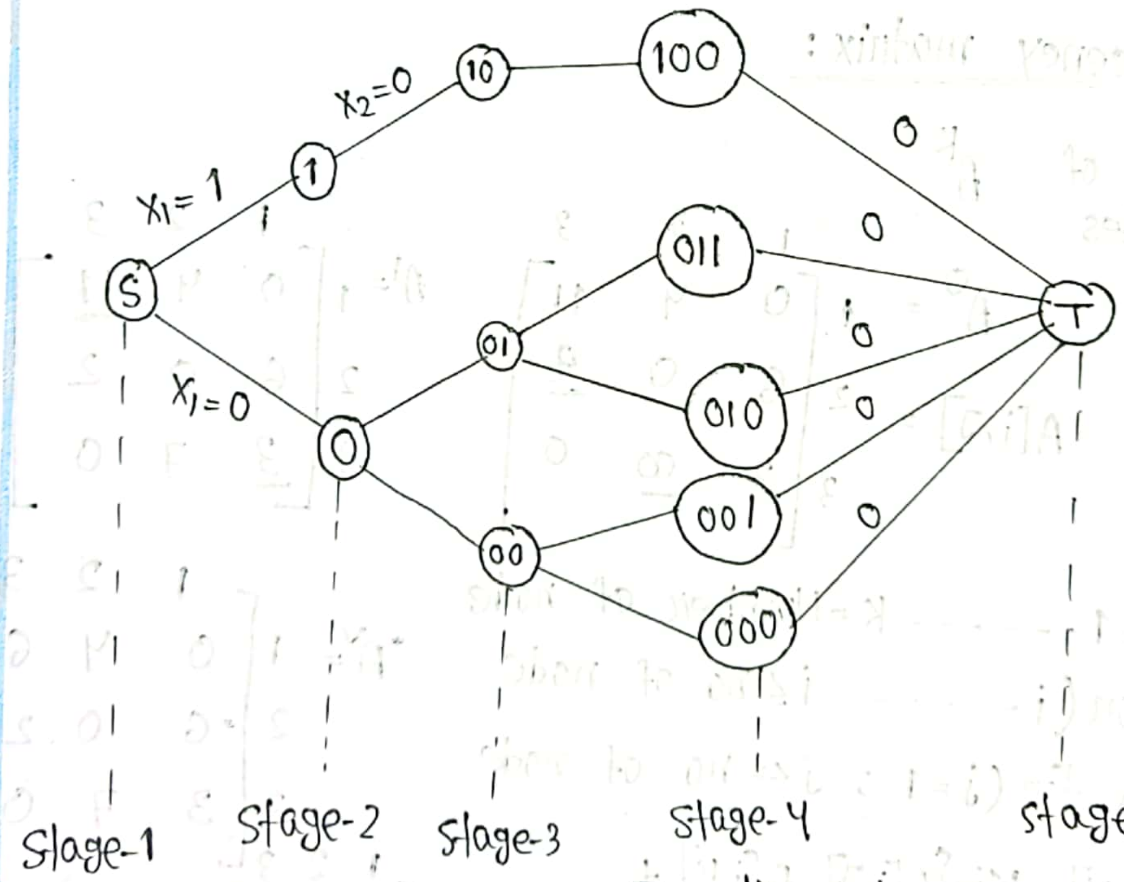
4.1 0/1 Knapsack Problem:

i	w_i	f_i
1	10	40
2	3	20
3	5	30

$x_i \to 0 \sim 1$

$x_i \to 0/1$

$M = 10$



$x_1 = 0$
$x_2 = 1$
$x_3 = 1$

$x_i = \{0, 1, 1\}$
Profit = 50

Fractional:
$x_i = (1/5, 1, 1)$
Profit = 58

Forward/Backward করে value করা লাগবে।

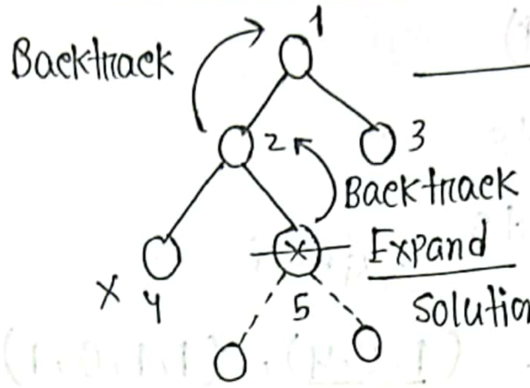
Lecture-10

TOPIC NAME: Backtracking Strategy

DAY: Wednesday

TIME:

DATE: 18/10/23



→ Solution state space tree

Solution probability = 0

[E] Problem : Sum of Subset -

n distinct positive number/ weight

combination → total weight = m
Sum of subset

$n=4$, $(w_1, w_2, w_3, w_4) = (11, 13, 24, 7)$

$m = 31$

Solution 1 : $(11, 13, 7)$ — 3 (w_1 w_2 w_4)

Solution 2 : $(24, 7)$ — 2 (w_3 w_4)

} Variable touple representation

$x_i \in w_i$ → Variable touple

Fixed Touple representation :

$x_i \in \{0, 1\}$

Solution 1 : $(1, 1, 0, 1)$

Solution 2 : $(0, 0, 1, 1)$

Constraints : (***)

① Implicit constraints

② Explicit constraints

Explicit constraints: $\overset{\nearrow x_i \in \omega_i}{(\underline{1,2,4})}$, $(1,1,0,1)$

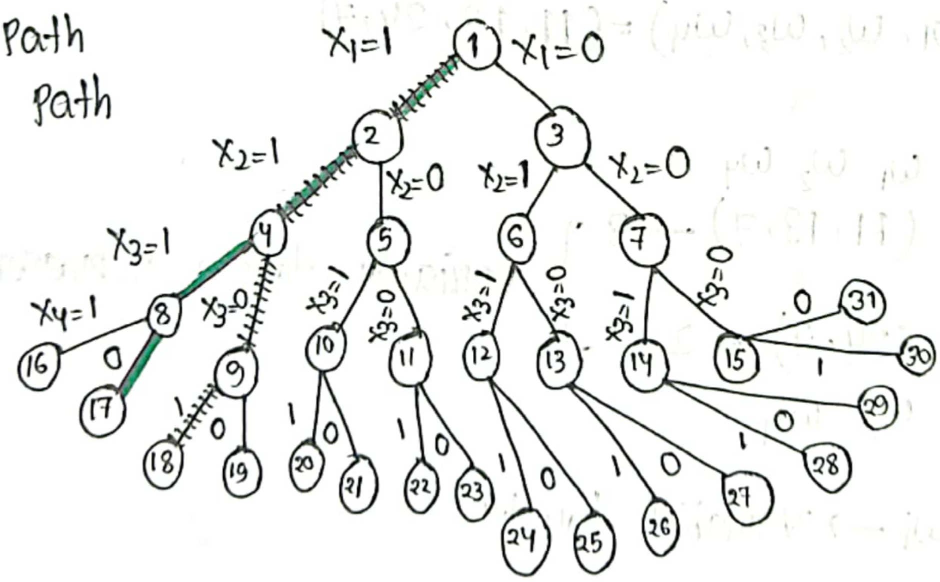
ω_i এর index $\qquad x_i \in \{0,1\}$

Implicit constraints: Summation = m

State space tree (Fixed Touple) : Numbering → BFS/DFS

||||| → Path

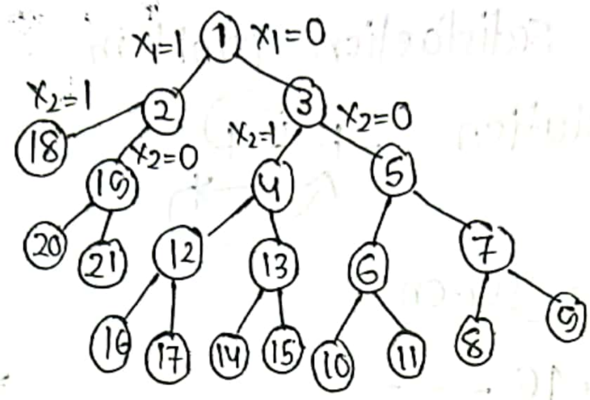
Solution path



$2^4 = 2^n$

(BFS) (1,1,0,1)

TOPIC NAME : DAY : TIME : DATE : / /

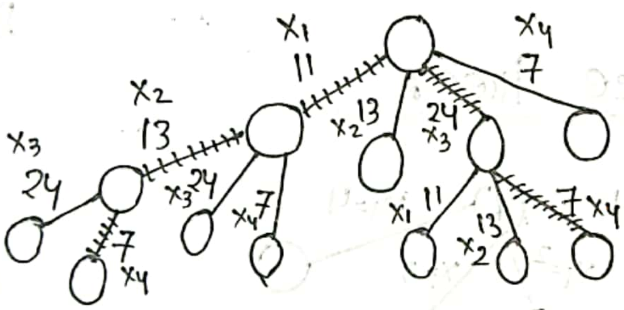


(DFS)

Solution space tree(variable Touple):

$\underline{1,2,3,4}$

weight indiees



(1,2,4)

(3,4) → weight index

Pintu Sir

Lecture-11

TOPIC NAME : Backtracking

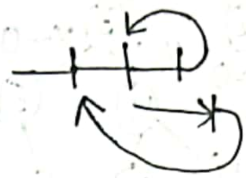
DAY : Tuesday

TIME :

DATE : 31 / 10

→ Constraints satisfaction problem

→ Partial solution



Problem: N - Queen -

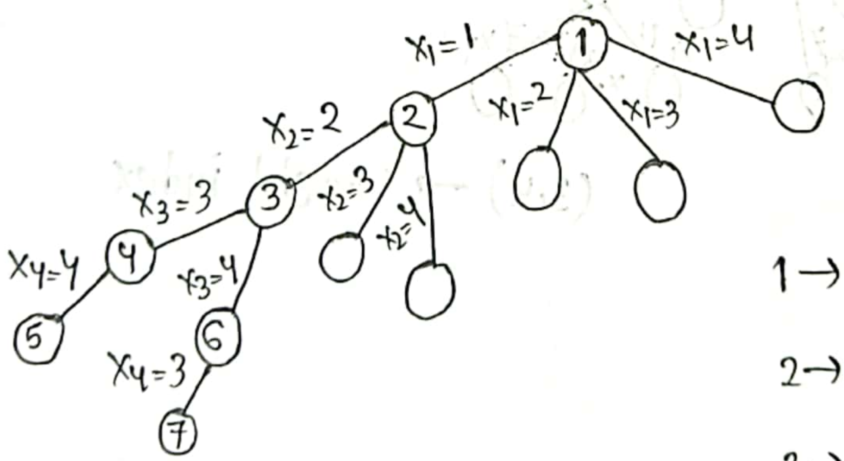
N = 4, 8, 16 - - - -

* Row, col, Diagonal বরাবর attack করবে।

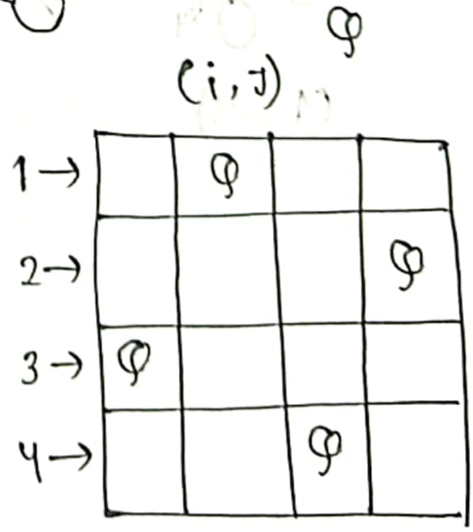
Constraints ⊏ Explicit → col No

⇒ check → col, Diagonal.

4- Queen state space tree:



(i, j)



4- Queen:

solution 2টা

① (2, 4, 1, 3) → col vector

TOPIC NAME :   DAY :
TIME :   DATE :   /   /

Algorithm :

① Row dedicated তাই attack করবে না।

② J এর value same হবে না।

③ Diagonal check :

$$|row_1 - row_2| == |col_1 - col_2|$$

↳ Diagonal attack

(*** Algorithm টা অনেক লাগবে।)

(***) What is N-queen prob ?,
N-queen implicit, explicit constraints,
Algorithm, State space tree.

Pintu sir

Lecture - 12

TOPIC NAME : Dynamic Programming DAY: Tuesday

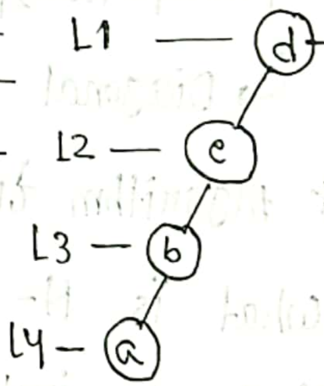
TIME : DATE : 7 / 11 / 2

Problem : Optimal Binary Search Tree ; → Must exam এ আসবে

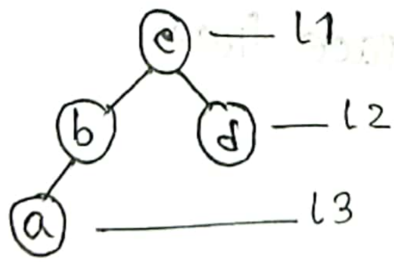
a_1, a_2, a_3 - - - - - - - - - - - - , a_m → list of items

Key $(a_K) = a_K$ → Probability of the items.

	a	b	c	d
	1	2	3	4
	0.1	0.2	0.3	0.4



cost → 1*0.4 + 2*0.3 + 3*0.2 + 4*0.1 = 2.0



cost = 1*0.3 + 2*0.4 + 2*0.2 + 3* 0.1 = 1.8

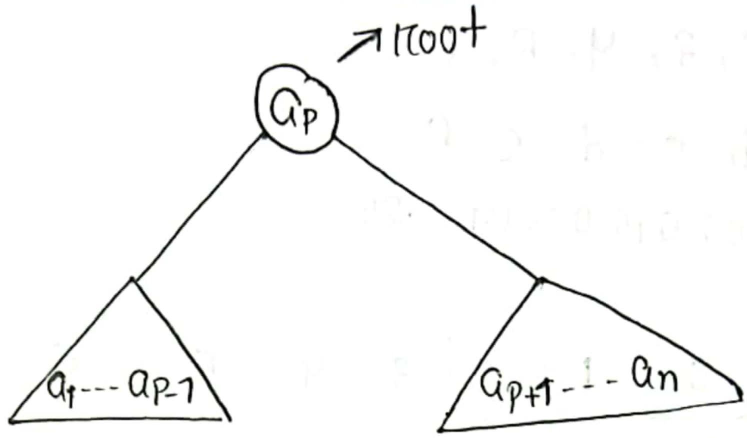
Algorithm / logic :

a b c d

1st root (c)

2nd root (b)

TOPIC NAME : DAY : TIME : DATE : / /



Root কে হবে =>

$$A[J,K] = \min_{J \le P \le K} \left(A[J,P-1] + \sum_{q=J}^{P-1} P(a_q) + A[P+1,K] + \sum_{q=P+1}^{K} P(a_q) + P(a_p) \right]$$

J to K এর মধ্যে কে root হবে, তা বের করা লাগবে

$$= \min_{J \le P \le K} \left(A[J,P-1] + A[P+1,K] + \sum_{q=J}^{K} P(a_q) \right) \quad —(1)$$

$$A[J,J] = P[a_j]$$

1, 2, 3

$\downarrow$ $P-1$ $\downarrow$ $P+1$

$1-1$ $= 4$

$=0$ => তাই matrix এর order এর খেয়াল নিতে হবে।

K= 1, 2, 3, 4, 5, 6

a b c d e f

0.4 0.05 0.15 0.05 0.1 0.25

	0	1	2	3	4	5	6	7
0	0							2.1
1	0	0.4	0.5					
2	0	0	0.05					
3	0	0	0	0.15				
4	0	0	0	0	0.05			
5	0	0	0	0	0	0.1		
6	0	0	0	0	0	0	0.25	
7	0	0	0	0	0	0	0	0

root matrix

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

$\sum_{q=J}^{K} P(a_q)$

TOPIC NAME :

DAY :

TIME : DATE : / /

	1	2	3	4	5	6
1	0.4	0.45	0.6	0.65	0.75	1
2						
3						
4						
5						
6						

Putting $P = 1, 2$ in ①

$$A[J,K] = \text{Min}\left(A[J] + A[2,K] + \sum_{q=j}^{K} P(a_q)\right)$$

$J = 1, K = 2$ $= 0.5$ for $1 = P$

$A[J,K] = 0.85$ for $2 = P$

A

2, 3

$$A[J,K] = \text{Min}\left[A[J, P-1] + A[P+1, K] + \sum_{q=j}^{K} P(a_q)\right]$$

$P=2$ $= \text{Min}\left[A[2,1] + A[3,3] + \sum_{q=2}^{3} P(a_q)\right]$

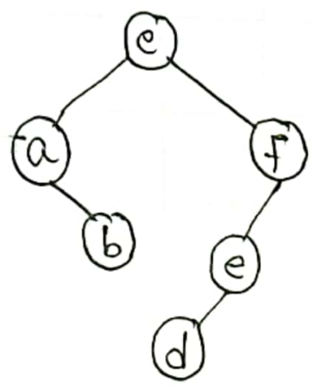
$= 0 + 0.15 +$

TOPIC NAME : DAY : TIME : DATE :

$$P=3 \quad = \min_{J \le P \le K} \left[A[J, P-1] + A[P+1, K] + \sum_{q=j}^{K} P(a_q) \right]$$

$$A[J, P-1] \downarrow 2,2 \qquad A[P+1, K] \downarrow 4,3$$

$$= 0.25$$



Lecture-13

TOPIC NAME: SSS Path

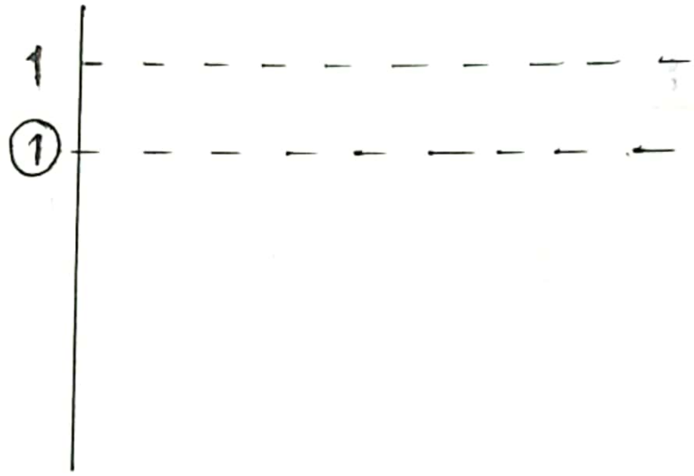
DAY: Wednesday

TIME:

DATE: 15/11/23

Single source shortest Path problem:

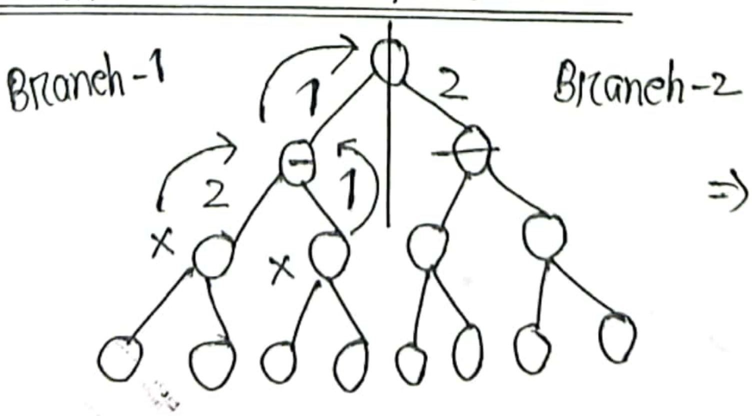
⇒ Bellman Ford Algorithm (Table কীভাবে update হচ্ছে, row by row দেখাতে হবে।)



Branch and Bound

Searching strategies

2-Queen → state space tree: Back tracking



⇒ Bound আসে = Optimal Solution এর Boundary থেকে হবে। এই Bound এ না থাকলে ঐ Branch এ যাব না।

⇒ Bound → Upper Bound $\underline{\text{Optimal Soln}}$ lower Bound

Problem: → 0/1

① Knapsack with B & B
② Traveling Salesman with B & B

যেকোন একটা exam এ আসবে।

0/1 Knapsack:

TSP:

#littleGiant-34