6/3/23

Knapsack   Dp approach

0/1 knapsack problem

weights : {3, 4, 6, 5}
profits : {2, 3, 1, 4}

W → represents weight of the bag.

$W = 8$

$n = 4$

w_i, P_i } weight, profit of corresponding items.

P_i	w_i	i \ W	0	1	2	3	4	5	6	7	8
		0	0	0	0	0	0	0	0	0	0
2	3	1	0	0	0	2	2	2	2	2	2
3	4	2	0	0	0	2	3	3	3	5	5
4	5	3	0	0	0	2	3	4	4	5	6
1	6	4	0	0	0	2	3	4	4	5	6

Write weights in ascending order of ~~profits~~

$\max(3+0, 2)$

Solution Set: { . . . . }

$i = \{1, 2, 3, 4\}$

Ans: {1, 0, 0, 1} ⇒ Solution.

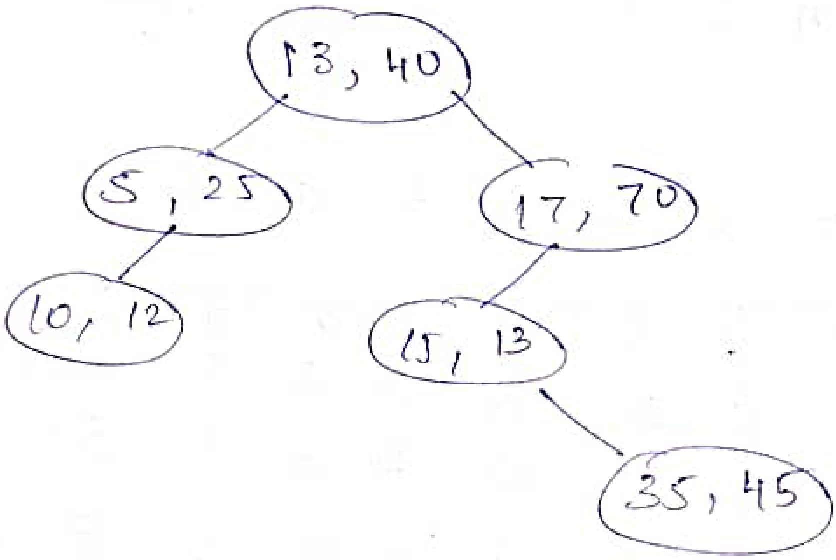
13/3/23

K-D Tree

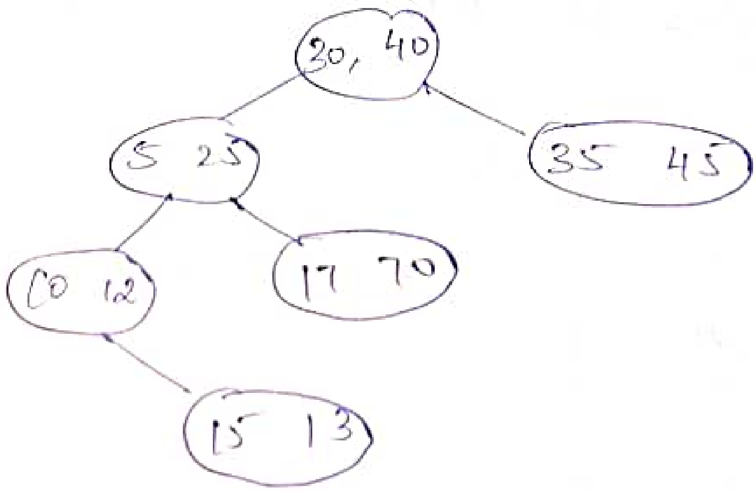
Space Partitioning tree

(13, 40) (5, 25) (10, 12)
(17, 70) (15, 13) (35, 45)

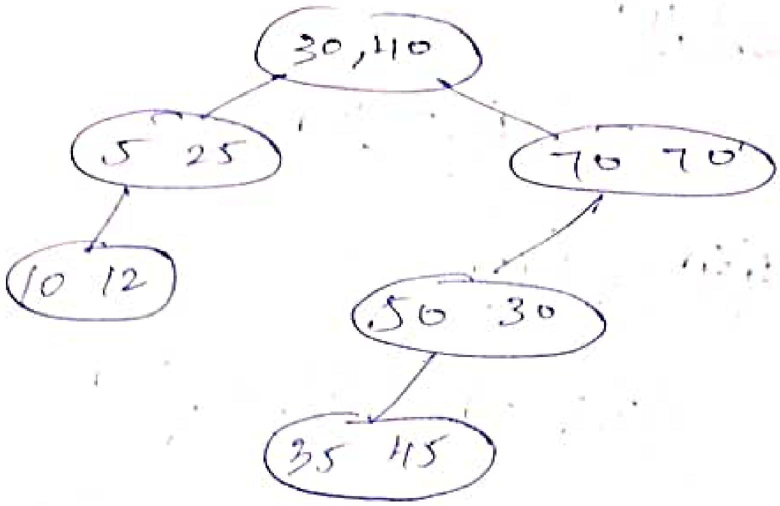
(12 ~~(15, 13)~~ ~~(35, 45)~~

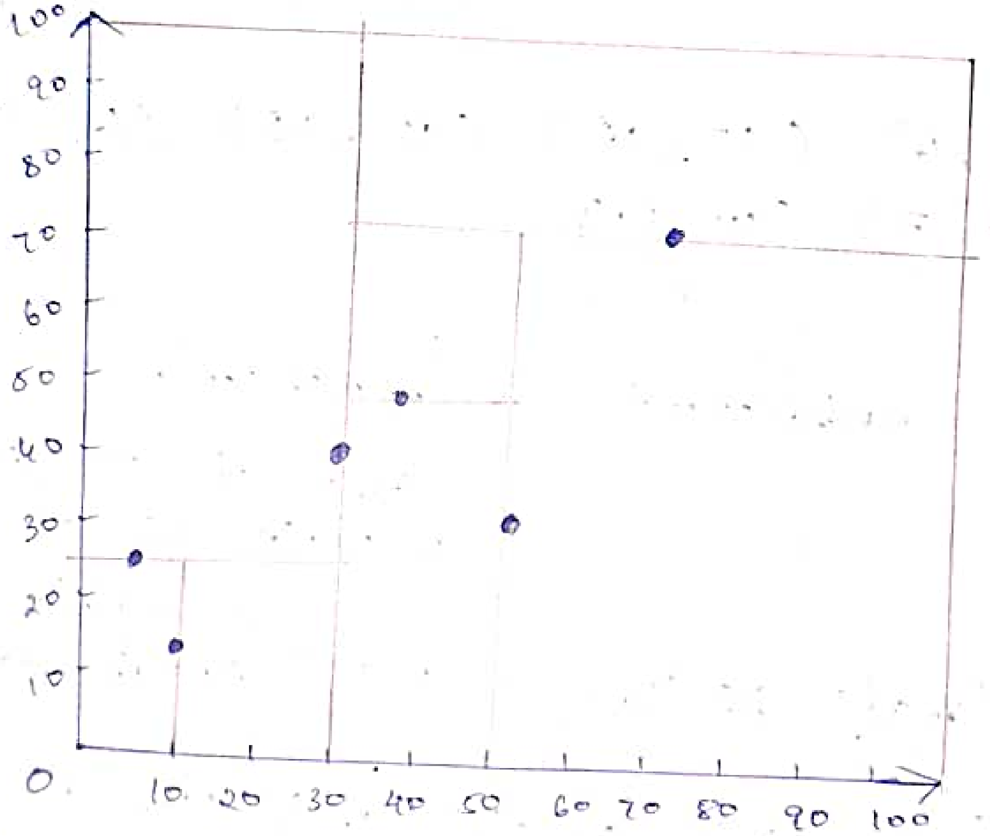


(30, 40) (5, 25) (10, 12) (17, 70) (15, 13)
(35, 45)



(30,40) (5,25) (10,12) (70,70) (50,30)
(35,45)

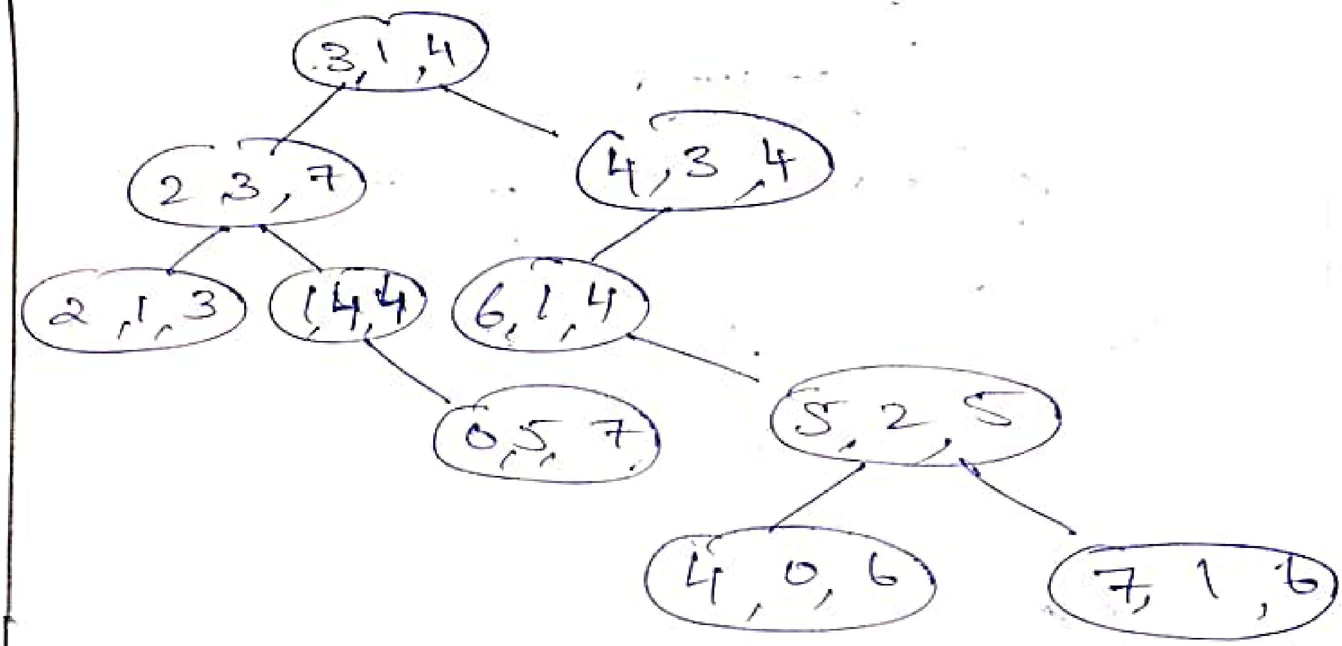




3D Tree:

(3, 1, 4) (2, 1, 3) (0, 5, 7)
(2, 3, 7) (6, 1, 4) (5, 2, 5)
(4, 3, 4) (1, 4, 4) (4, 0, 6)
(7, 1, 6)

Construct 3D tree



(3, 6) (17, 15) (13, 15) (6, 12) (9, 1)
(2, 7) (10, 19)

unbalanced

balanced.

sort in x direction.
middle in root

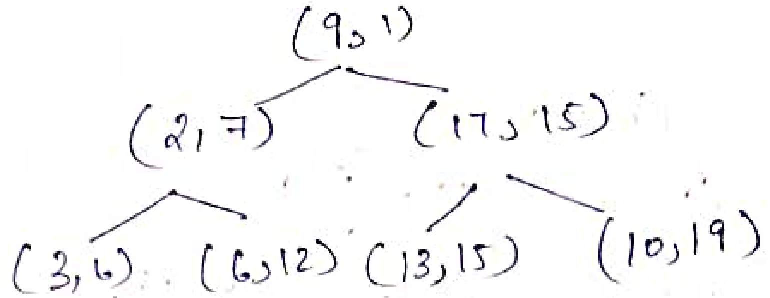
Balanced KD Tree: → searching operation → static data points

(2, 7) (3, 6) (6, 12) (9, 1) (10, 19) (13, 15)
(17, 15)

(9, 1)

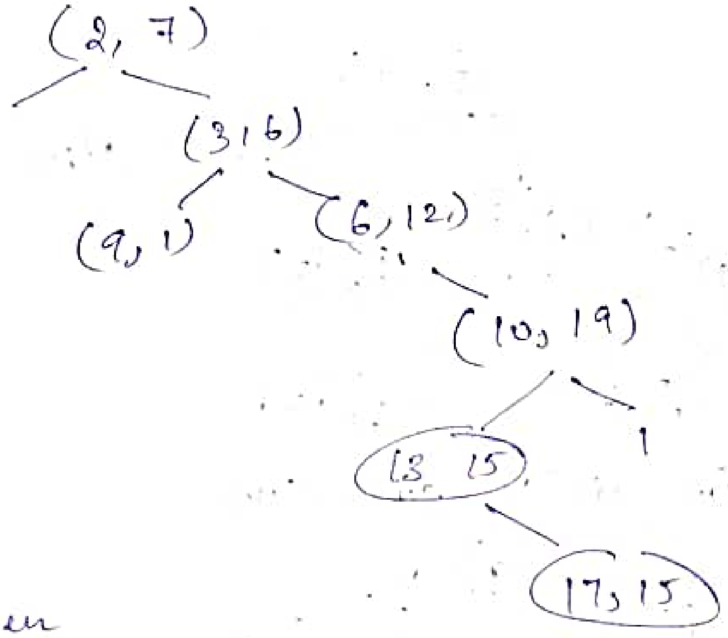
(3, 6) (2, 7) (6, 12)

(13, 15) (17, 15) (10, 19



Unbalanced:

H = 4



Recommended

N - even

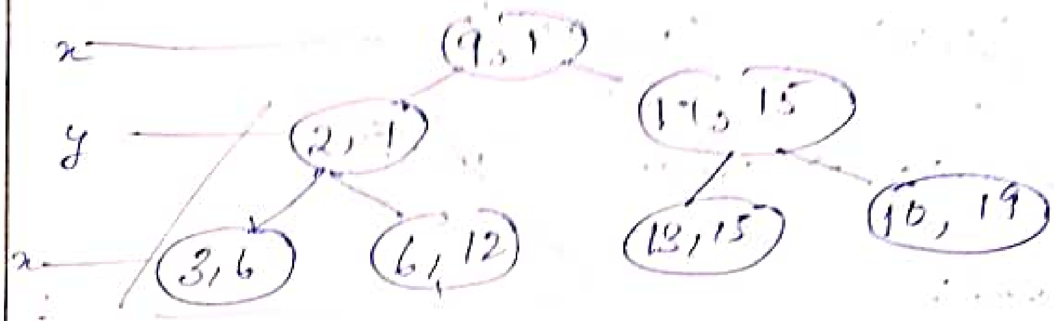
- $\lfloor \frac{N}{2} \rfloor$ on left
- $\lfloor \frac{N}{2} \rfloor + 1$ th => root or
- others => right.

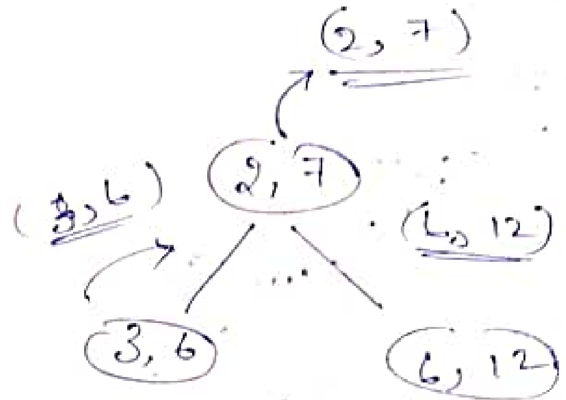
Min in x direction:

if Dimension of current node == given
min → left.
left subtree → empty
=> return node value.

else. min ⟨ left, right } recursive. ; current

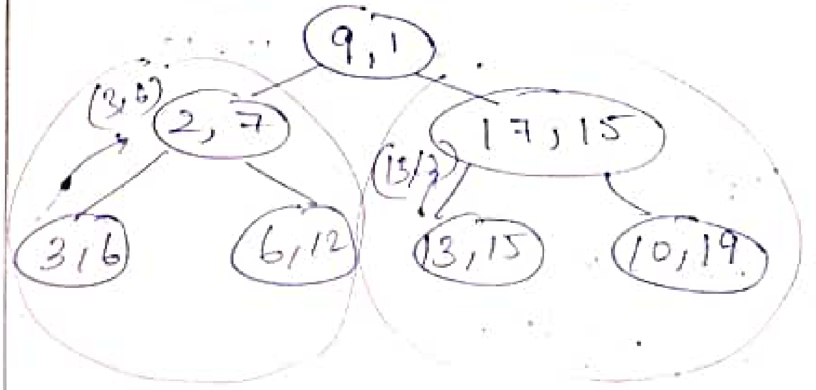
min { —, —, — }





Min = (2, 7)

Min in y direction



$$\min \left\{ (3, 6) \quad (9, 1) \quad (13, 15) \right\}$$

[y direction]

→ (9, 1)

1) Deletion:

* Current Node has the data:

1.1) * leaf Node → simply delete

1.2) * right subtree.

1.3) * left subtree.

1.4) * both left and right

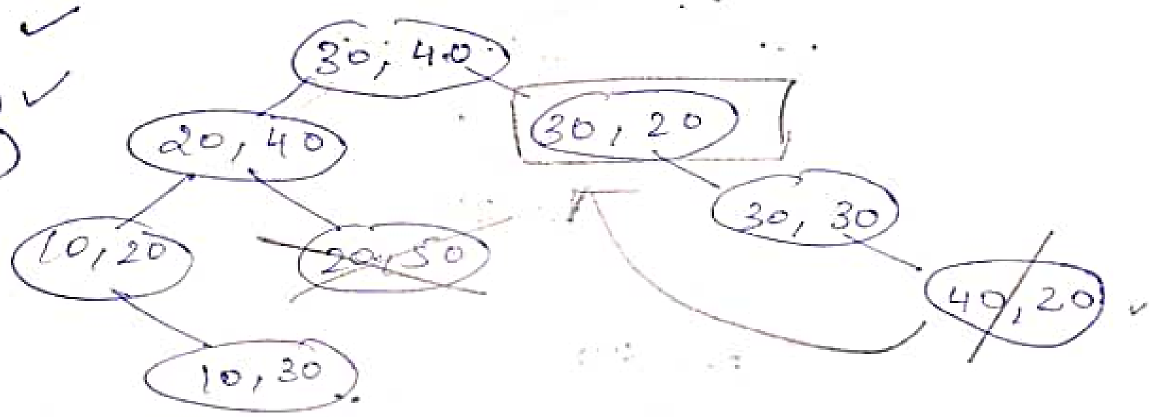
right subtree

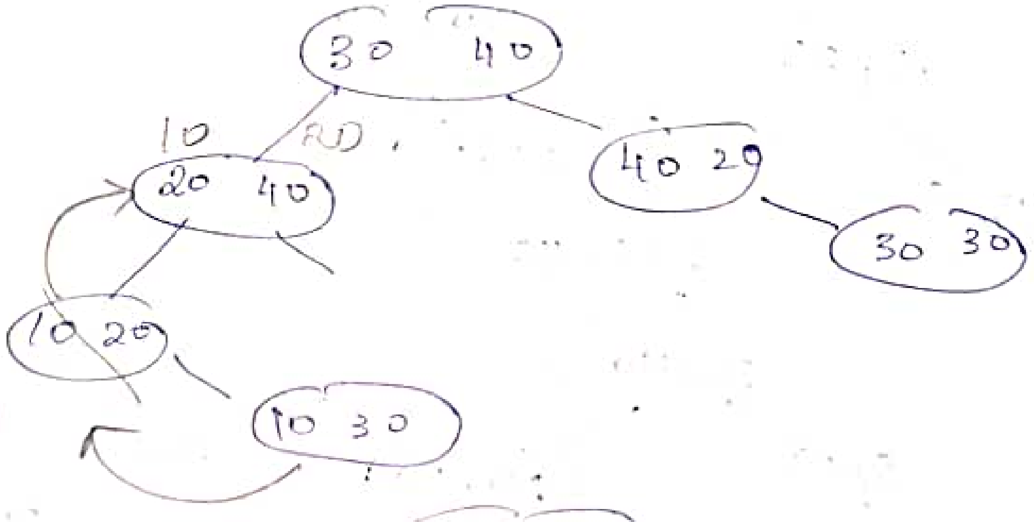
1.2.1) min of current node's dimension,
1.2.2) replace and recursively do

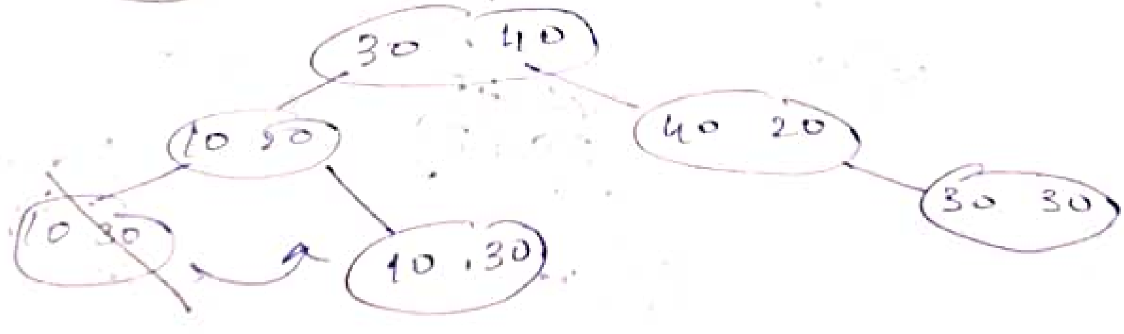
left subtree :
1.3.1) ~~max~~ min of current node's dimension.
1.3.2) Replace and Recursively
1.3.3) new left subtree.

both
do 1.2)

Q) (20,50) ✓
(30,20) ✓
(20,40)

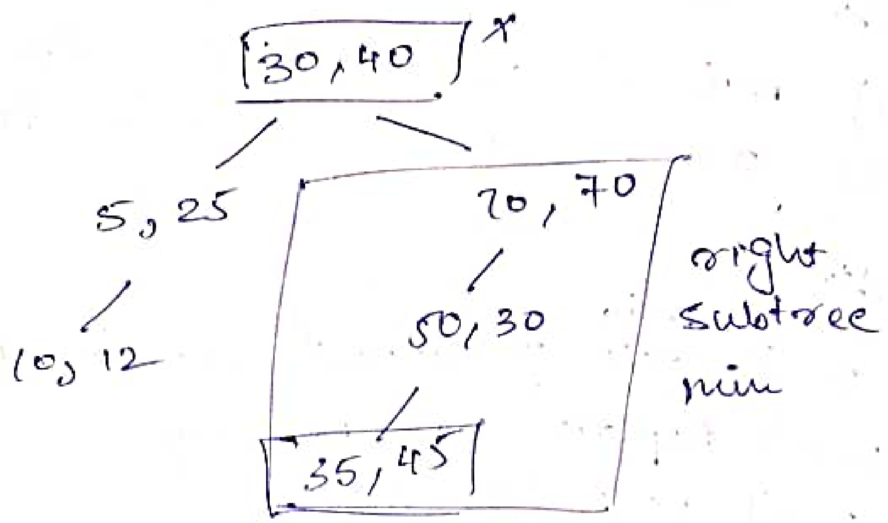


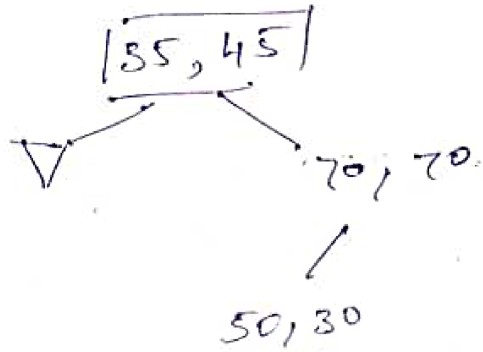


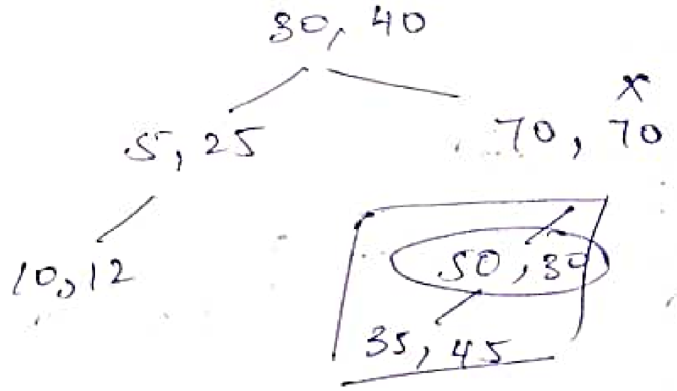


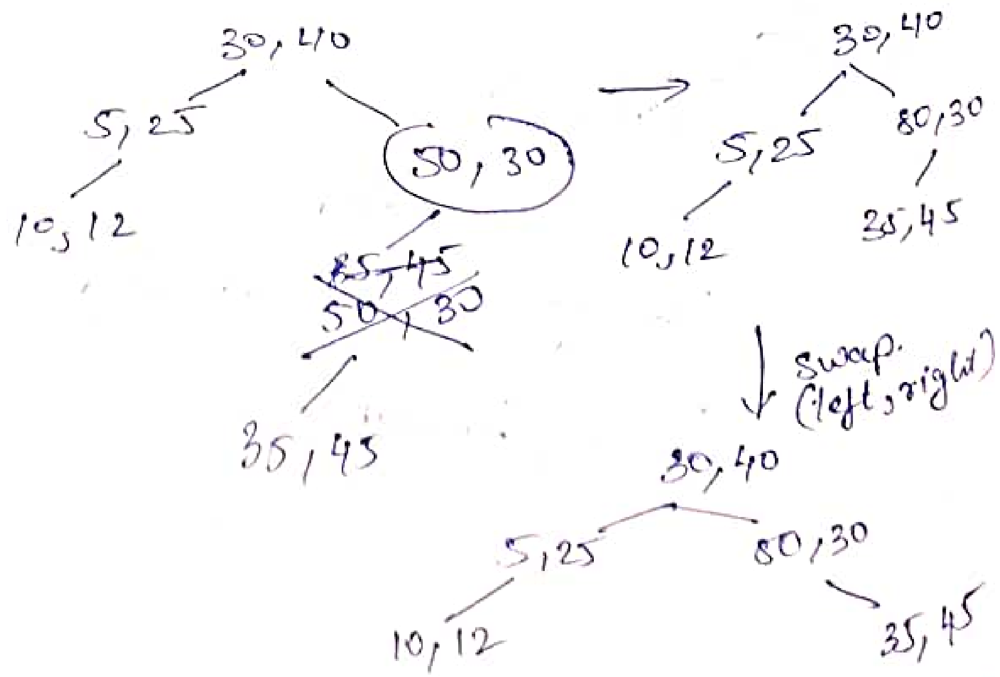
Q) delete (30, 40)

15/3

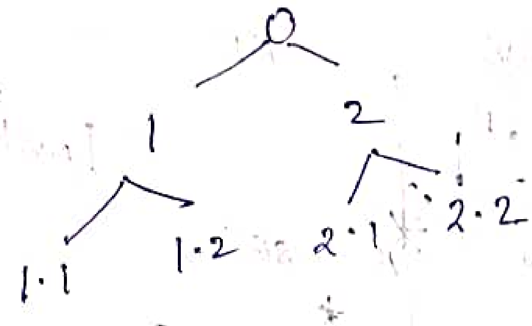


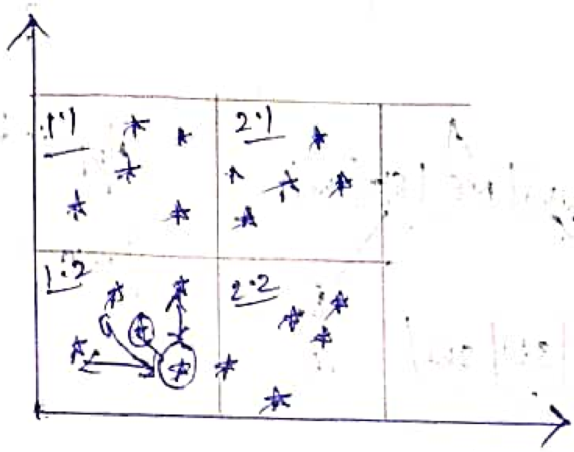




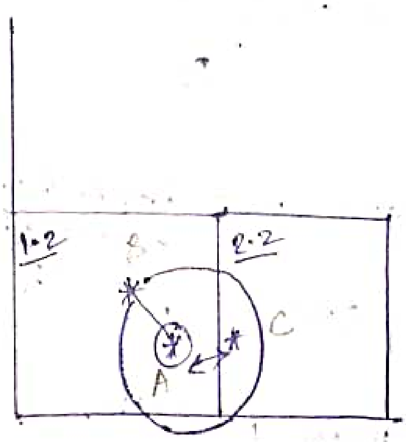


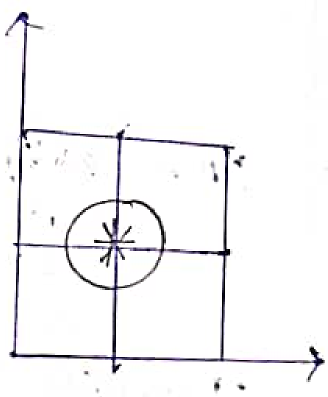
KNN Nearest Neighbour





Worst Case





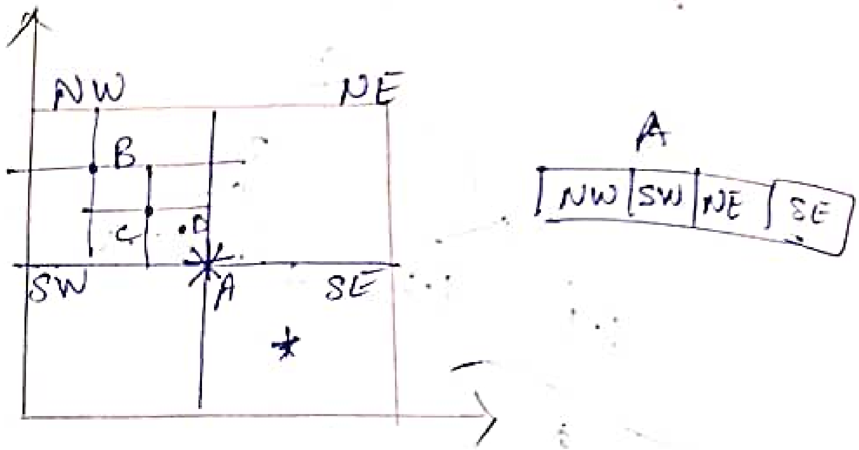
Quad Tree

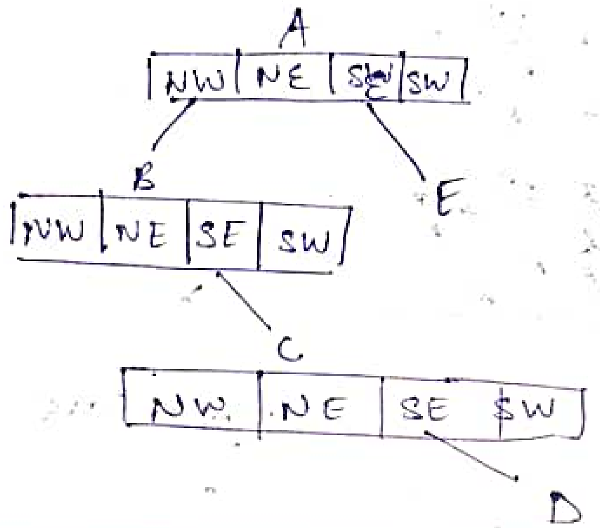
→ 0/4 children.

→ 2D grid.

KD Tree → 2D or any higher dimension.

Quad Tree





Application

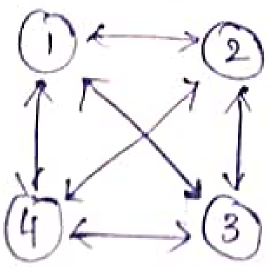
Image Processing.

---

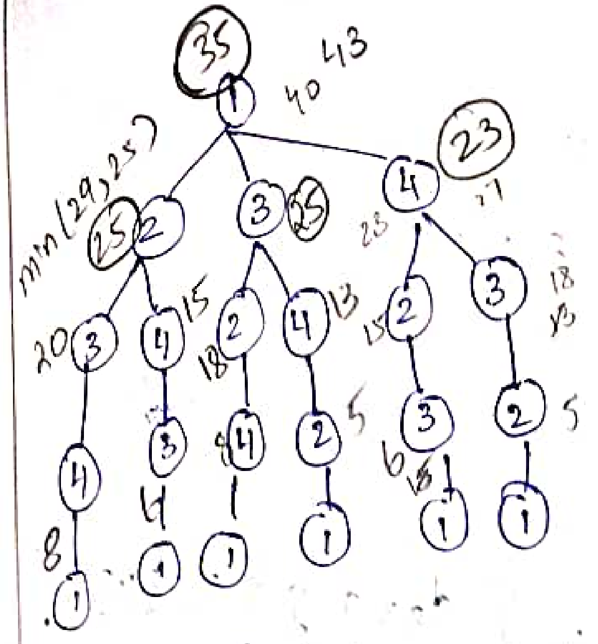
* DP – subproblems → dependent on each other
  ↳ ~~add~~ store results.

* <u>Travelling Salesman problem.</u>
* Matrix Chain Multiplication.
* Optimal Binary Tree.

17/3/23



$$
\begin{array}{c}
 \\ 1 \\ 2 \\ 3 \\ 4
\end{array}
\begin{array}{c}
\begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\begin{bmatrix}
0 & 10 & 15 & 20 \\
5 & 0 & 9 & 10 \\
6 & 13 & 0 & 12 \\
8 & 8 & 9 & 0
\end{bmatrix}
\end{array}
$$



Route:

1 ⟶ 2 ⟶ 4 ⟶ 3 ⟶ 1

35

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

$$= \min_{k \in \{2,3,4\}} \left\{ \cancel{dd} \cdot d_{1k} + g\{k, \{2,3,4\} - \{k\}\} \right\}$$

Top down

$$g\{i, S\} = \min_{k \in \{S\}} \left\{ d_{ik} + g\{k, \{S\} - \{k\}\} \right\}$$

Bottom Up:

$g(4, \phi) = 8$

$g(3, \phi) = 6$

$g(2, \phi) = 5$

$$g\{2, \{3\}\} = d_{23} + g(3, \phi) = 9 + 6 = 15$$

$$g\{2, \{4\}\} = d_{24} + g(4, \phi) = 10 + 8 = 18$$

$g\{3, \{2\}\} = 13 + 5 = 18$

$g\{3, \{4\}\} = 12 + 8 = 20$

$g\{4, \{2\}\} = 13$

$g\{4, \{3\}\} = 15$

$$g\{2, \{3,4\}\} = \min \begin{Bmatrix} d_{23} + g\{3, \{4\}\}, \\ d_{24} + g\{4, \{3\}\} \end{Bmatrix}$$

$$= \min\{29, 25\}$$

$$= 25$$

$$g\{3, \{2,4\}\} = \min \begin{Bmatrix} d_{32} + g\{2, \{4\}\}, \\ d_{34} + g\{4, \{2\}\} \end{Bmatrix}$$

$$= 25$$

$$g\{4, \{2,3\}\} = 23$$

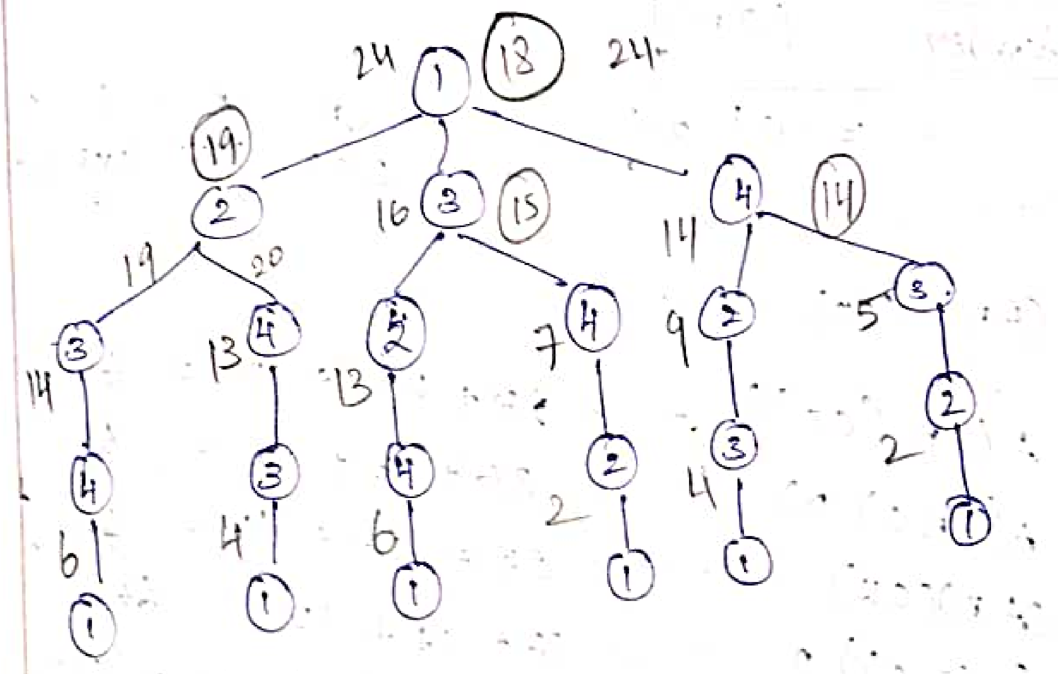
$$g\{1, \{2,3,4\}\}$$

$$= \min \begin{bmatrix} d_{12} + g\{2, \{3,4\}\} \\ d_{13} + g\{3, \{2,4\}\} \\ d_{14} + g\{4, \{2,3\}\} \end{bmatrix}$$

$$= 35, 40, 43$$

$$= \underline{35}$$

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & & & & \\ 2 & & & & \\ 3 & & & & \\ 4 & & & & \end{array} \quad \begin{bmatrix} 0 & 5 & 3 & 10 \\ 2 & 0 & 5 & 7 \\ 4 & 3 & 0 & 8 \\ 6 & 5 & 9 & 0 \end{bmatrix}$$



Route:

$1 \longrightarrow 3 \longrightarrow 4 \longrightarrow 2 \longrightarrow 1$

Cost: $3 + 8 + 5 + 2 = 18$

Matrix Chain Multiplication

ABC

A	B	=	C
5×4	4×3		5×3

$5 \times 3 \times 4 = 15 \times 4$

$= \underline{60}$

A	B	C
2×3	3×4	4×5

$(AB)C$: $24 + 40 = 64$

$A(BC)$: 90

35 23 24 45

BC : 60 + 30

A(BC) $60 + 30 = 90$ 23 30 $\boxed{90}$

A	B	C	D
2×3	3×4	4×3	3×2

$\frac{5 \cdot 6}{6} = 5$ $\qquad \frac{(2 \cdot 3)!}{4!\,3!}$

Catalan Number $\quad \frac{(2n)!}{(n+1)!\; n!}$

A	B	C	D
2×3	3×4	4×3	3×2

(AB) (CD)

A(B(CD)) ⟶ 24 + 24 + 12 = 60 ✓
A((BC) D) ⟶ 36 + 18 + 12 = 66
(AB)(CD) ⟶ 24 + 24 + 16 = 64
(A(BC)) D ⟶ 36 + 18 + 12 = 66
((AB) C) D. ⟶ 24 + 24 + 12 = 60 ✓

20/8/23

Matrix chain Multiplication

A_1	A_2	A_3	A_4
5×4	4×6	6×2	2×7

5 binary trees.

5·4
6!
3! 4!
= 5

$A_1 \mid (A_2 A_3)$

M

	1	2	3	4
1	0	120	88	158
2		0	48	104
3			0	84
4				0

S

	1	2	3	4
1	0	1	1	3
2		0	2	3
3			0	3
4				0

$M[1,2]$

$= 5 \times 4 \times 6$

$= 120$

$M[2,3]$ $\quad$ $A_1 \mid A_2$

$= 4 \times 6 \times 2 = 48$

$M[3,4] = 6 \times 2 \times 7$ $\quad$ $\underline{M[1,4]}$

minimum cost.

$M[1\ 3] =$

$A_1\ A_2\ A_3$

$5 \times 4 \quad 4 \times 2$

$5 \times 0 \quad 4 \times 3$

$(A_1 A_2) A_3 \qquad A_1 (A_2 A_3)$

$$M[1,3] = \min \begin{cases} M[1,1] + M[2,3] + 40 \\ M[1,2] + M[3,3] + \cancel{30}\ 60 \end{cases}$$

$= \{0 + 48 + 40,\ 120 + 0 + 60\}$

$= \min\{88, 180\} = 88.$

$A_2 (A_3 A_4)$

$M[2,4] = M[2,2] + M[3,4] + 4 \times 6 \times 7$

$= 0 + 84 + 168 = 252$

$= M[2,3] + M[4,4] + 4 \times 2 \times 7$

$= 48 + 0 + 56 = 104$

$A_1 A_2 A_3 A_4$

$M[1,4]$

$= M[1,1] + M[2,4] + 5 \times 4 \times 7$

$= 0 + 104 + 140 = 244$

$= M[1,2] + M[3,4] + 5 \times 6 \times 7$

$= 120 + 84 + 210 = 414$

$M[1,3] + M[4,4] + 5 \times 2 \times 7$

$= 88 + 0 + 70 = \underline{158}$

Min Cost: 158.

$$(A_1)\,|\,(A_2 \quad A_3)\,|\,(A_4)$$

Split up: look at S matrix.

$$(A_1 (A_2 A_3)) A_4$$

Q)

A	B	C	D
4×5	5×3	3×2	2×7

$$
\begin{array}{c|c}
M & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} &
\begin{bmatrix} 0 & 60 & 70 & 126 \\ - & 0 & 30 & 100 \\ - & - & 0 & 42 \\ - & - & - & 0 \end{bmatrix}
\end{array}
\qquad
\begin{array}{c|c}
S & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} &
\begin{bmatrix} 0 & 1 & 1 & 3 \\ - & 0 & 2 & 3 \\ - & - & 0 & 3 \\ - & - & - & 0 \end{bmatrix}
\end{array}
$$

$M[1,2] = 4 \times 5 \times 3 = 60$

$M[2,3] = 5 \times 3 \times 2 = 30$

$M[3,4] = 3 \times 2 \times 7 = 42$

$M[1,3] = \cancel{M[1,2] + M[2,3]}$

$\cancel{M[1,3] +}$

$M[1,1] + M[2,3] + 4 \times 5 \times 2$
$= 0 + 30 + 40 = 70$

$M[1,2] + M[3,3] + 4 \times 3 \times 2$
$= 60 + 0 + 24 = 84$

$M[2,4]$

$= M[2\,2] + M[3\,4] + 5 \times 3 \times 7$

$= 0 + 42 + 105 = 147$

$= M[2\,3] + M[4\,4] + 5 \times 2 \times 7$

$= 30 + 0 + 70 = 100$

$M[1\,4]$

$M[1\,1] + M[2\,4] + 4 \times 5 \times 7$

$= 0 + 100 + 140 = 240$

$M[1\,2] + M[3\,4] + 4 \times 3 \times 7$

$= 60 + 42 + 84 = 186$

$M[1\,3] + M[4\,4] + 4 \times 2 \times 7$

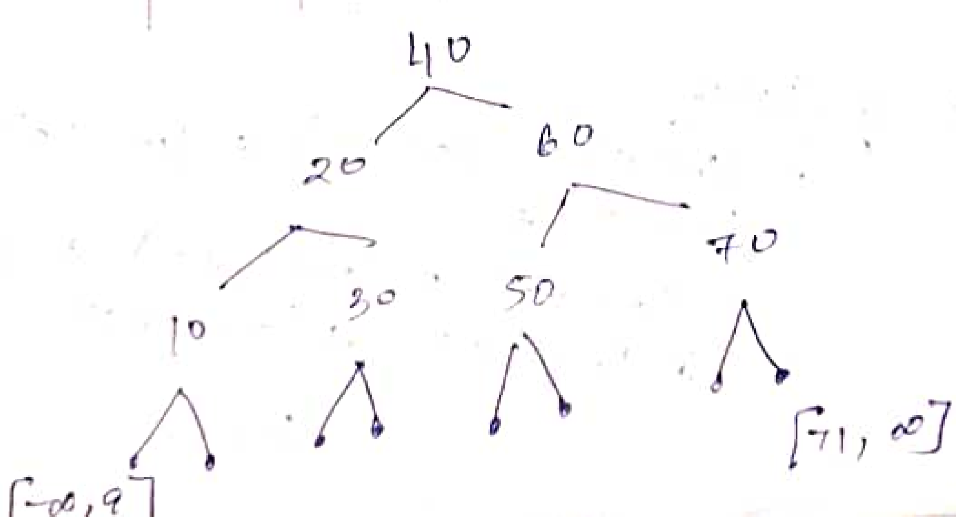
$= 70 + 0 + 56 = 126.$

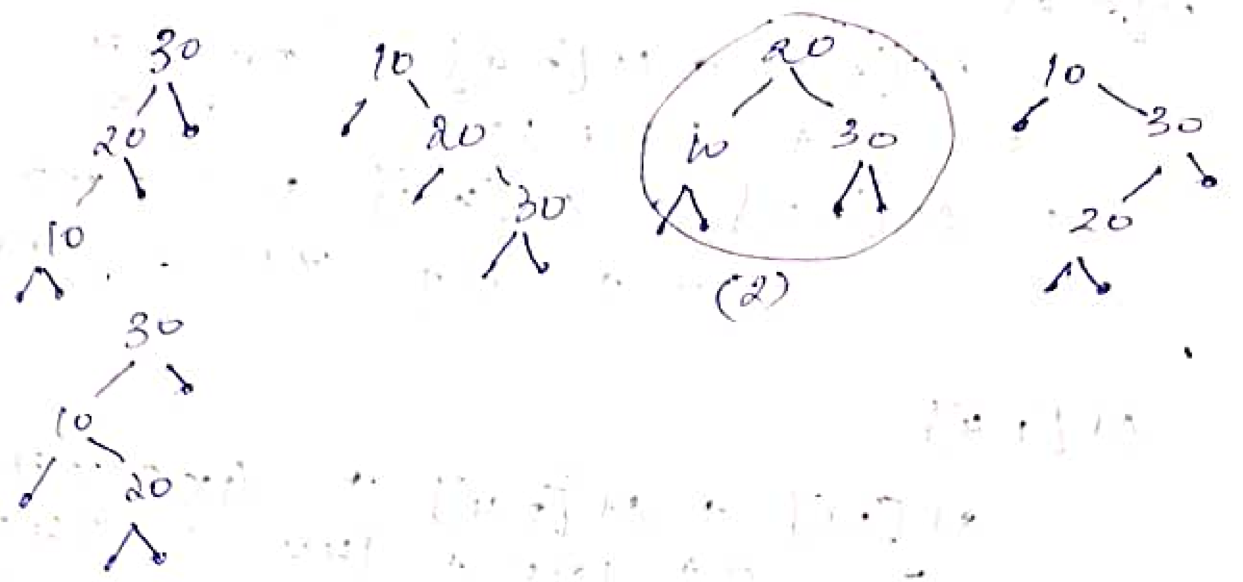
Split Up:

$((A)(BC))(D)$

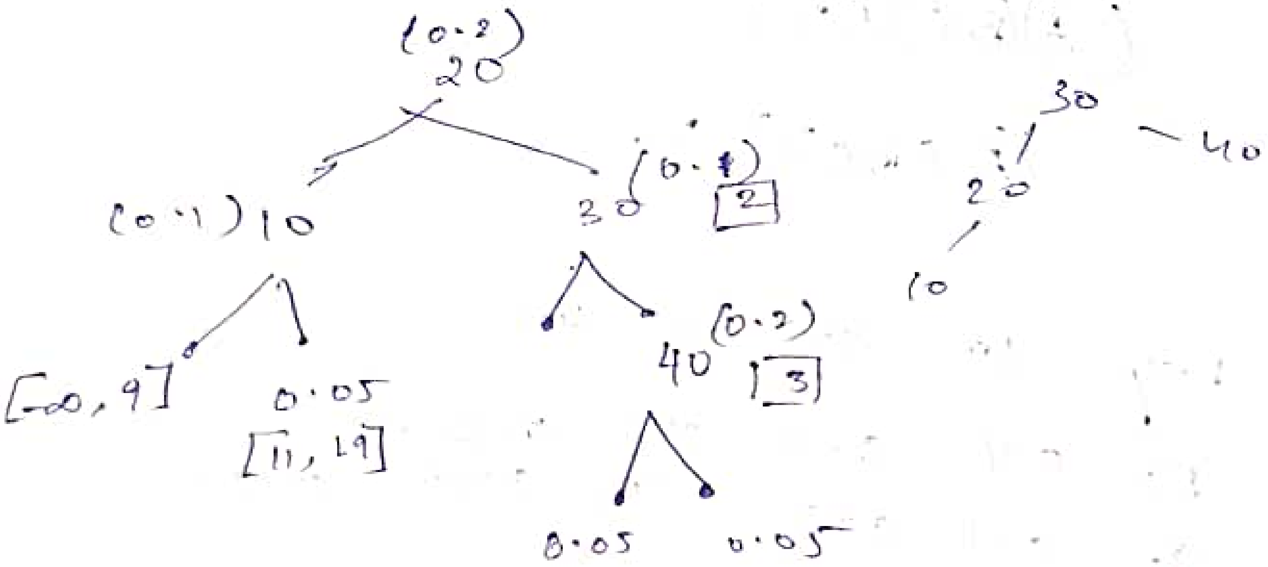
$(A(BC))D.$

key	10	20	30	40	
p_i	0.1	0.2	0.1	0.2	
q_i	0.1	0.05	0.15	0.05	0.05



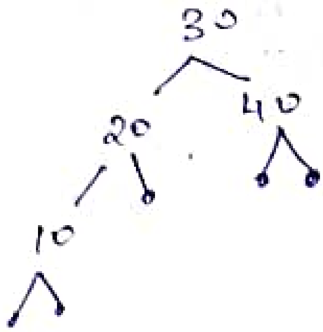


key :	10	20	30	40	
p_i :	0.1	0.2	0.1	0.2	
q_i :	~~0.05~~ 0.1	0.05	0.15	0.05	0.05



Cost of Search

$= (0.2 \times 1) + (0.1 \times 2) + (0.1 \times 2) +$
$(0.2 \times 3) +$
$(0.1 \times 2) + (0.05 \times 2) + (0.15 \times 2)$
$+ (0.05 \times 3) + (0.05 \times 3)$
$= 2.1.$



27/3/23

Optimal BT

$$C[i,j] = \min_{i<k\le j} \left\{ \begin{array}{l} c[i,k-1] + \\ c[k,j] \end{array} \right\} + w(i,j)$$

$r[i,j]$

$w(i,j) = w(i,j-1) + p_j + q_j$

$c[i,i] = 0$

$w(i,i) = q_i$

$r[i,i] = 0$

diff
$j-i$

$w(0,0) = q_0$

$w(0,1) = w(0,0) + p_1 + q_1$

p_1

$$C[0,3] = \min_{0<k\le 3} \left\{ \begin{array}{l} c[0,0] + c[1,3], \\ c[0,1] + c[2,3], \\ c[0,2] + c[3,3] \end{array} \right\} + w(0,3)$$

$r[i,j] =$ k where min occured.

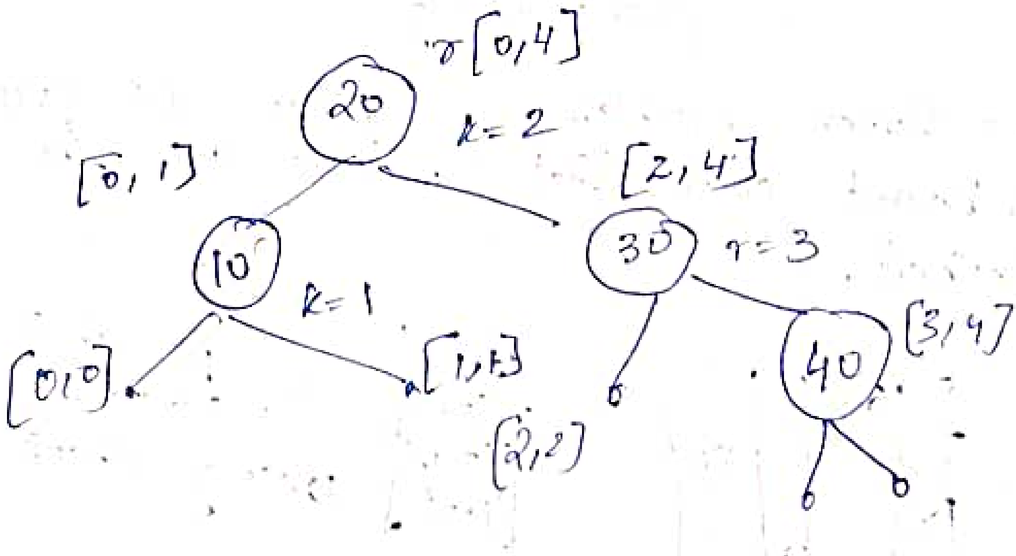
diff	
$j-i=0$	$c[0,0]$ $c[1,1]$ $c[2,2]$ $c[3,3]$
$j-i=1$	$c[0,1]$ $c[1,2]$ $c[2,3]$
$j-i=2$	$c[0,2]$ $c[1,3]$
$j-i=3$	$c[0,3]$

	1	2	3	4
Key:	10	20	30	40
p:	3	3	1	1

q: 2 3 1 1 1

j-i=0	c[0,0] = 0 w(0,0) = 2 r[0,0] = 0	c(1,1) = 0 w(1,1) = 3 r(1,1) = 0	c(2,2) = 0 w(2,2) = 1 r(2,2) = 0	c(3,3) = 0 w(3,3) = 1 r(3,3) = 0	c(4,4) = 0 w(4,4) = 1 r(4,4) = 0
j-i=1	w(0,1) = 8 c(0,1) = 8 r(0,1) = 1	w(1,2) = 7 c(1,2) = 7 r(1,2) = 2	w(2,3) = 3 c(2,3) = 3 r(2,3) = 3	w(3,4) = 3 c(3,4) = 3 r(3,4) = 4	
j-i=2	w(0,2) = 12 c(0,2) = 19 r(0,2) = 1	w(1,3) = 9 c(1,3) = 12 r(1,3) = 2	w(2,4) = 5 c(2,4) = ~~8~~ 8 r(2,4) = 3		
j-i=3	w(0,3) = 14 c(0,3) = 25 r(0,3) = 2	w(1,4) = 11 c(1,4) = 19 r(1,4) = 2			
j-i=4	w(0,4) = 16 c(0,4) = 32 r(0,4) = 2				

w(0,4) = w(0,3) + 2



9]

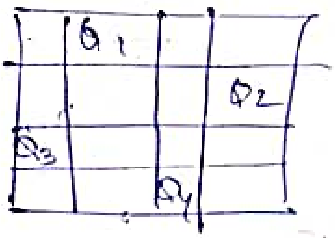
Key:	1	2	3	4	5
p :	0.15	0.1	0.05	~~0.0~~ 0.1	0.2.

q: 0.05, 0.1, 0.05, 0.05, 0.05, 0.1

29/3/2023

Back tracking

N - Queen problem - place N queens in chess board such that nothing can be attacked.



(4,3)
(3,1)
(2,4)
(1,2)

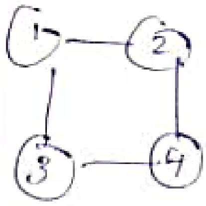
(1 possible solution)

using stack.

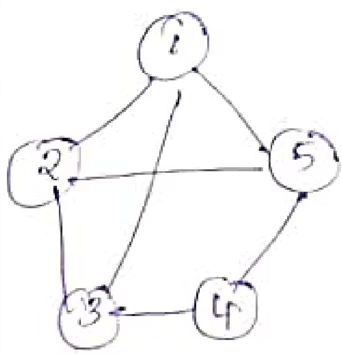
Graph Coloring

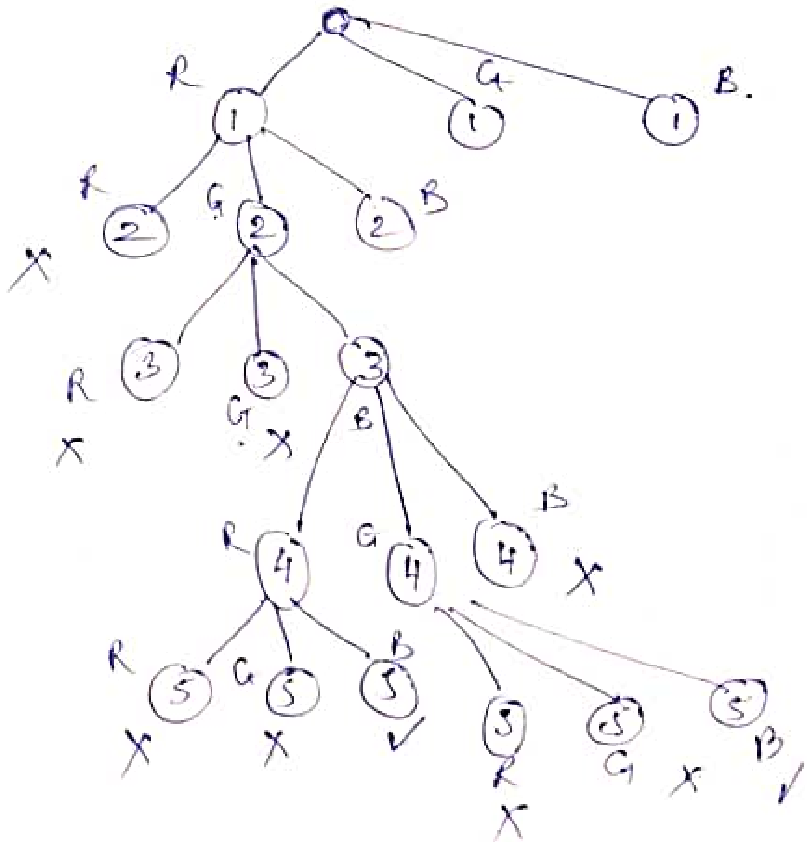
Decision | Optimization

State Space tree



H/W





1	2	3	4	5
R	G	B	R	B
R	G	B	G	B

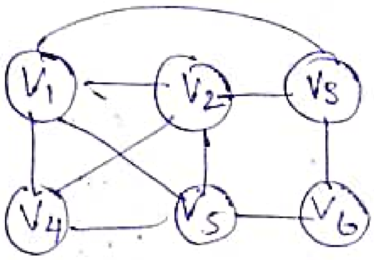
$$
\begin{array}{c|ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & 0 & 1 & 1 & 0 & 1 \\
2 & 1 & 0 & 1 & 1 & 1 \\
3 & 1 & 1 & 0 & 1 & 0 \\
4 & 0 & 1 & 1 & 0 & 1 \\
5 & 1 & 1 & 0 & 1 & 0
\end{array}
$$

Hamiltonian cycle: all vertice visited exactly once except start vertex.

all cycls

TSP: only one cycle

— Connected graph.

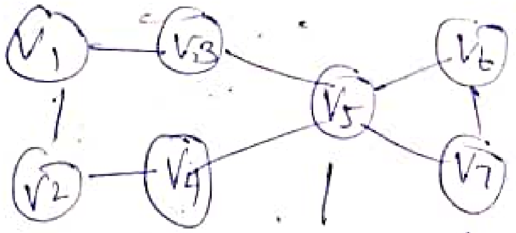


$V_1\ V_2\ V_3\ V_6\ V_5\ V_4\ V_1$

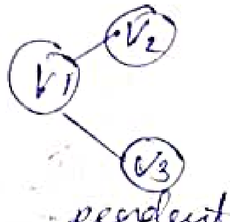
V_1 ~~V2~~ $V_3\ V_6\ V_5\ V_4 V_2 V_1$

$V_2\ V_3\ V_6\ V_5\ V_4\ V_1\ V_2$

changing only starting point → Same H.C



Cut vertex
articulation pt.



pendent vertex

not possible to have H.C.

$$
\begin{array}{c|ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & 0 & 1 & 1 & 0 & 1 \\
2 & 1 & 0 & 1 & 1 & 1 \\
3 & 1 & 1 & 0 & 1 & 0 \\
4 & 0 & 1 & 1 & 0 & 1 \\
5 & 1 & 1 & 0 & 1 & 0
\end{array}
$$

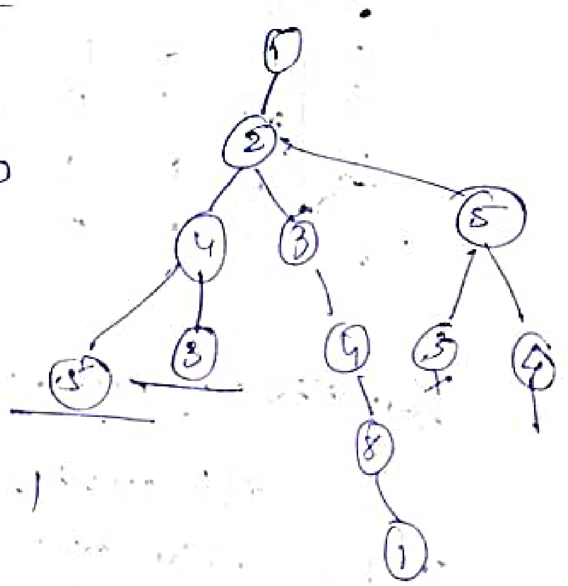
1-2-3-4-5-1

1	2	3	4	5
1	2	3	4	5

* vertex not repeated
* connection from prev. vertex forward vertex

Condition.

1	2	3	4	5
1	2	3	4	0
1	2	3	0	0
1	2	~~3~~ 4	~~3~~ 5	
1	2	4	~~5~~ 0	0
1	2	5	~~3~~ 4	3	1

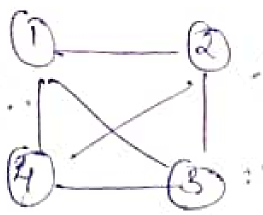


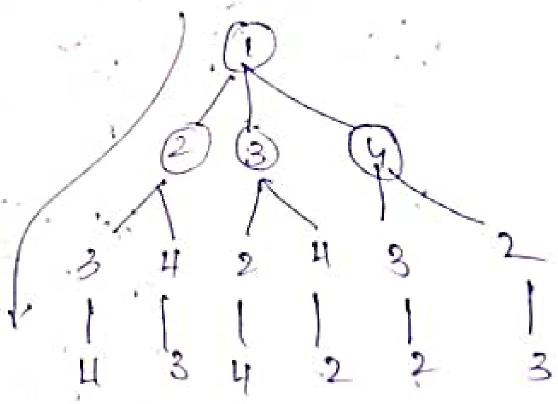
$\boxed{1\ 2\ 5\ 4\ 3\ 1}$

Branch and bound → FIFO
→ LIFO
→ LC. Least Cost

→ We use BFS (Backtracking : DFS)

eg: TSP:





LIFO

4
3
2

2	3	4	3	4

↑
FIFO

(Queue) → BFS / level order traversal.

We use only least cost branch & bound.

$$\begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix} \begin{array}{c} \text{min} \\ 10 \\ 2 \\ 2 \\ 3 \\ 4 \\ \hline 21 \end{array} \Rightarrow$$

Reduced

$$\begin{bmatrix} \infty & 10 & 20 & 0 & 1 \\ 13 & \infty & 14 & 2 & 0 \\ 1 & 3 & \infty & 0 & 2 \\ 16 & 3 & 15 & \infty & 0 \\ 12 & 0 & 3 & 12 & \infty \end{bmatrix}$$

$$\begin{array}{ccccc} 1 & 0 & 3 & 0 & 0 \end{array}$$

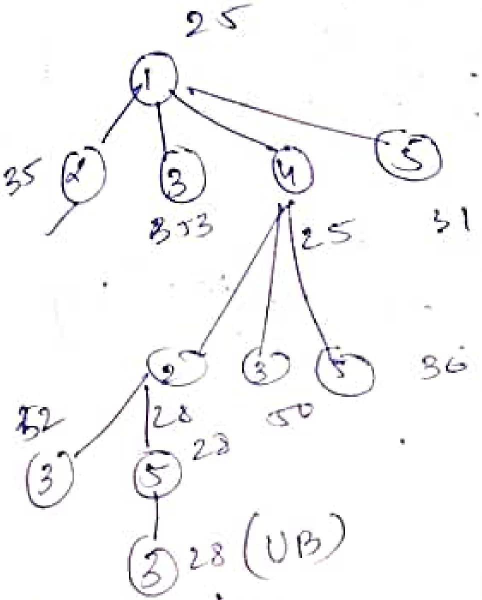
Cost = 4

Total reduction Cost

= 25

$\Downarrow$

$$\begin{bmatrix} \infty & 10 & 17 & 0 & 1 \\ 12 & \infty & 11 & 2 & 0 \\ 0 & 3 & \infty & 0 & 2 \\ 15 & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & 12 & \infty \end{bmatrix}$$



LC Branch and bound.

r c
[1, 2] → change row 1 and Column 2 to ∞.
2 → 1 is ∞.

New matrix

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 11 & 2 & 0 \\ 0 & \infty & \infty & 0 & 2 \\ 15 & \infty & 12 & \infty & 0 \\ 11 & \infty & 0 & 12 & \infty \end{bmatrix}$$

If all entry ∞ no need of 0.

Reduced matrix
Cost of reduction = 0

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 12 & \infty & 11 & \infty & 0 \\ 0 & 3 & \infty & \infty & 2 \\ \infty & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & \infty & \infty \end{bmatrix}$$

[1, 4] ✓ → min in 2nd level.

For 3rd level → use this matrix

[4, 5]

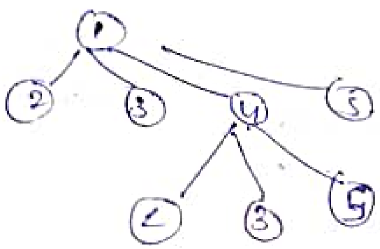
$$\begin{array}{c c}
\begin{bmatrix}
\infty & \infty & \infty & \infty & \infty \\
12 & \infty & 11 & 2 & \infty \\
0 & 3 & \infty & 0 & \infty \\
15 & 3 & 12 & \infty & \infty \\
11 & 0 & 0 & 12 & \infty
\end{bmatrix} &
\begin{array}{c} \\ 2 \\ 3 \\ \\ \\ \end{array}
\end{array}$$

(5) $25 + 5 + 1$

$= 31$

As [1, 4] → least.

New matrix
$$\begin{bmatrix}
\infty & \infty & \infty & \infty & \infty \\
12 & \infty & 11 & \infty & 0 \\
0 & 3 & \infty & \infty & 2 \\
\infty & 3 & 12 & \infty & 0 \\
11 & 0 & 0 & \infty & \infty
\end{bmatrix}$$

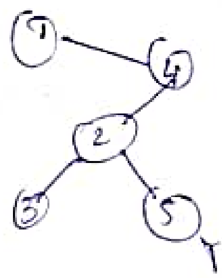


Now again,
check each vertex by making it as [4, 2].
Take min out.

[4, 2]

[4, 3]

[4,5]

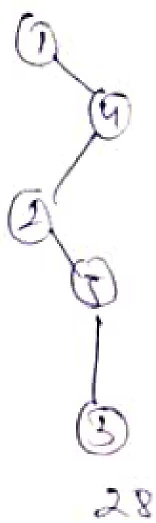


min

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 12 & \infty & 11 & \infty & 0 \\ 0 & \infty & \infty & \infty & 2 \\ \infty & \infty & \infty & \infty & \infty \\ 11 & \infty & 0 & \infty & \infty \end{bmatrix}$$

[2,3]

[2,5]



28

Min Cost = 28

Upper bound = ∞ till completion of lvl 1.

Note:
Lower and Upper bound are needed for knapsack

3/4

LC Branch & Bound
(0/1 knapsack)

Capacity : 15

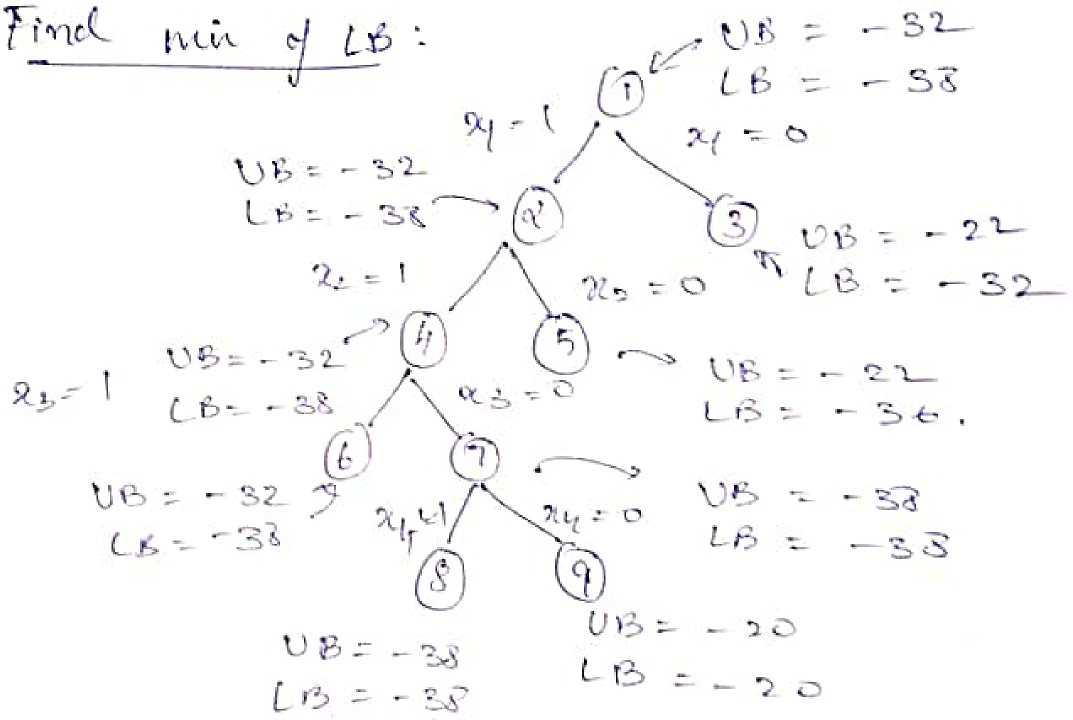
Item	profit	weight
1	10	2
2	10	4
3	12	6
4	18	9

UB = 10 + 10 + 12 = 32

LB = $10 + 10 + 12 + \frac{3}{9} \times 18$

= 38.

Find min of LB :



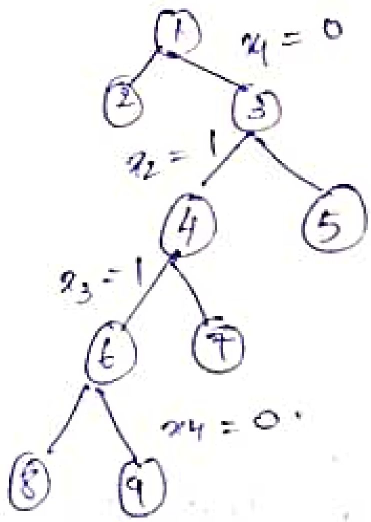
Max profit = 38 (by LB)

Solution. = 1 1 0 1

x_1 x_2 x_3 x_4.

6)

Item	Profit	Weight
1	24	24
2	18	10
3	18	10
4	10	7



0 1 1 0

Max profit: 36.

Capacity : 25

String Matching.

S | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | $l_s = 10$

? ?

$l_p = 4$

P | 1 | 2 | 3 | 4 |

Max no. of shifts $= l_s - l_p + 1$
(Comparison)

```
for (int i = 1 ; i <= max ; i++ )
{   flag = TRUE
    for (j = 1 ; j <= lp && flag ; j++ )
    {       if (p[j] == s[j+i-1])
                flag = FALSE ;
    }
}
```

Algo 1:

i

S 1 2 3 5 4 1 3 2

P 1 2 3 4

j

14/3

S: A B C D A B C E

P: B C E

2 3 5 = 10

Valid hit

hash function ⟶ simple addition.

⟶ needs to be simple ⟶ constant time

Rolling hash function ⟶ on previous value find hash from prev. hash value.

6 − 1 + 4 = 9

9 − 2 + 1 = 8.

S: c c a c c a a c d b a.

P: d b a = 7

(*) hash value matching with parent ⟶ check the character, then come to a conclusion.

Spurious hit ⟶ Both hash values are matching, characters not matching

Valid hit ⟶ Both hash, char matching

Not possible to completely eliminate spurious hit.

P ⟶ m length   S ⟶ n

$= P[1] \times 10^{m-1} + P[2] \times 10^{m-2} + \dots P[m] \times 10^{0}$

P: d b a $= 4 \times 10^2 + 2 \times 10 + 1 = 421$

4 2 1

S: c c a c c a a c d b a

$3 \times 10^2 + 3 \times 10^1 + 1 \times 10^0$

$= 331$

$[331 - 3 \times 10^2] \times 10 + 3$

$= 313$

Hash fn → more complex
→ avoid spurious hit

$n: 500$ (10 char)
$m: 20$

$$\left(P[1] \times 10^{19} + P[2] \times 10^{18} + \cdots \right) \bmod 32$$

Designing hash fn → to maximise spurious hit

KMP: Knuth - Morris - Pratt

a a b c a d a a b d b a
0 1 0 0 1 0 1 2 3

check for prefix and suffix in pattern

1 2 3 4 5 6 4 3

P: (1) (12) (123) (1234) (12345) . . . .
S: (643) (43) (3)

(3) (43) (643) (5643) (45643) (345643) (2345643)
(12345643)

LPS table → longest prefix same as suffix
↓
π table