Page :
Date :

Notations
complexity class
Recurrences

Selection, insertion sort.

divide & Conquer

merge sort
Quick sort
min max
Binary Search

Greedy

Dijkstra : single source shortest path
Knapsack fractional
job sequenc.
min cost spann.: kruskal + prim

Dynamic prog^n

- Multistage graphs
Bellman ford
0/1 Knapsack
- floyd warshell
- Assembly line sched^n.
TSP
- Longest common subseq^n

Backtrack^n Branch bound

TSP, 15 puzzle

N queen
+ of subsets
Graph coloring

String matching

Naïve
Robin karp
knuth morris pratt

Complexity.

PAGE ____
DATE: / /

Multistage graph $O(n + |E|)$.

SSSP : BF : $O(|V| - |E|)$

APSP : FW : $\Theta(n^3)$ ∵ 3 nested loops used.

ALS : $O(n)$.

✓ 0/1 K : $O(n * M)$.

TSP : $O(2^n \cdot n^2)$.

ALS n
NQu n^2
fw n^3

✓ LCS : ~~X~~ $O(mn)$.

✓ NQu. : $O(n^2)$.

ΣSubs. : $\sum_{n=0}^{n} 2^n$ Exp. order

Graph. : $O(n 2^n)$.

✓ Naive : $O(m \times n)$

MCSP : prim $O(n^2)$ can be reduced to $O[|E| \log|V|]$ using ds
: krus ~~$O[|E| \log |V|]$~~
$O[|E| \log |E|]$.

AB CD BACDE & CBAF   LCS : CBAF

MLNOM & MNOM   MNOM

ACBAEB

Naive string matching

Q.1 T = Planingand analysis   text

P = and.   pattern.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
t	p	l	a	n	i	n	g	a	n	d	a	n	a	l	y	s	i	s

	1	2	3
p	a	n	d

t_i	p_j		
1	1	Compare $t_i=1$ and $P_j=1$: p & a : ≠	t_i++
2	1	l ≠ a X t_i++	reset
3	1	✓	t_i+, P_j+
4	2	✓	t_i+ P_j+
5	3	X	t_i++, P_j=1
6	1	X	
7	1	X	
8	1	✓	
9	2	✓	
10	3	✓	

match found.

Q.2

	1	2	3	4	5	6
T =	a	c	a	a	b	c
P =	a	a	b			

1	1	✓
2	2	X
3	1	✓
4	2	✓
5	3	✓

Match found.

✓ : i++, j++.
X : i++, j=1.

Backtracking
Brute force approach

Q.3 T = 0 0 0 0 1 0 0 1 0 1 0 0 0 1
P = 0 0 0 1

1 1	✓	1 1	✓	4 1	✓
2 2	✓	2 2	✓	5 2	X
3 3	X	3 3	✓	3 1	
4 4	X	4 4	X		
5 1	✓	~~3 1~~	✓		
6 1	✓	~~4 2~~	✓		
7 2	✓	5 3			
8 3	✓	2 1	✓		
9 4	✓	3 2	✓		
		4 3	✓		
		5 4	✓		

1 match found.   1 Match found.

15 − 4 = 11 = no. of shifts

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	1	0	0	1	0	1	0	0	0	1	

shift		
0	0 0 0 1	stop
1	0 0 0 1	✓
2	0 0 0 1	stop
3	0 0 0 1	stop
4	0 0 0 1	stop
5	0 0 0 1	✓
6	0 0 0 1	X
7	0 0 0 1	X
8	0 0 0 1	X
9	0 0 0 1	X
10	0 0 0 1	✓
11		

Table of Contents

MODULE 2

Chapter 3 : Divide and Conquer   3-1 to 3-29

MODULE 3

Chapter 4 : Greedy Algorithms

MODULE

Chapter 5 : Dynamic Programming

AND.

18 − 3 = 15 shift.

Shift		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
		P	L	A	N	I	N	G	A	N	D	A	N	A	L	Y	S	I	S
0	×	A	N	D															
1	×		A	N	D														
2	×			A	N	D													
3	×				A	N	D												
4						A	N	D											
5							A	N	D										
6								A	N	D									
7	✓								A	N	D								
8										A	N	D							
9											A	N	D						
10												A	N	D					
11													A	N	D				
12														A	N	D			
13															A	N	D		
14																A	N	D	
15																	A	N	D

at i = 8, j = 1 till i = 10, j = 3.
shift 7.

it took 12 Comparisons to find a match.
↳ end tak ja match ke.

	1	2	3	4	5	6	7	8	9	10	11	12
	a	b	c	a	b	d	a	a	b	c	d	e
0	a	b	d									
		a										
			a									
✓				a	b	d						
					a							
						a						
							a	b				
								a	b	d		
									a			
										a		
											a	
												a.

it took 8 Comparisons to find a match.

Page :
Date :

Rabin karp algorithm.

Text $\overset{1}{a}\overset{2}{a}\overset{3}{a}\overset{4}{a}\overset{5}{a}\overset{6}{b}$

pattern $\overset{1}{a}\overset{1}{a}\overset{2}{b}$.

a	1
b	2
c	3
⋮	
j	10

10 alphabets.

$\underbrace{\overset{3}{c}\ \overset{2}{c}\ \overset{1}{a}}_{7}\ \overset{3}{c}\ \overset{3}{c}\ \overset{1}{a}\ \overset{1}{a}\ \overset{5}{e}\ \overset{4}{d}\ \overset{2}{b}\ \overset{1}{a}$

pattern $\underset{4\ 2\ 1}{dba} = 7$

now 7 is matching but $d \neq c$ ∴ Spurious hit

$O(mn)$ ∵ ~~Brute force~~

$O(n-m+1)$ if no spurious hits.

To avoid spurious hits, strong ~~Rabin karp~~ hash fun^n^ is used.

pattern d b a

$4\times10^2 + 2\times10^1 + 1\times10^0 = \underline{421}$

a
b
c
⋮
z — 26. $26^2\ 26^1$...

~~10~~ 26^3...

Now

$\overset{300+}{}$ $\underline{\underset{3\times10^2}{c}\ \underset{3\times10^1}{c}\ \underset{1\times10^0}{a}}\ c\ c\ a\ a\ e\ d\ b\ a = 331.$

a, A
b B
c C
⋮
z Z ... 127^2 127^1...

rolling hash fun^n^.

$$[[3\times10^2 + 3\times10^1 + 1\times10^0] - 3\times10^2] \times10 + 3\times10^0$$

$3\times10^2 + 3\times10^1 + 1\times10^0 =$ ~~421~~ $.331$

$331 - 300 = 31$, $31\times10 = 310$, $310+3 = 313$

$313 -$ $3\times10^2 + 1\times10^1 + 3\times10^0 = 300+10+3 = 313$

∴ TC

Camlin

acha work karta only if repeated chars in string.

Knuth-Morris-Pratt (KMP)

pattern : abc dabc

(to avoid no. of comparisons on the same letters).

prefix : a, ab, abc, abcd.

suffix : ~~c, cb, cba, c~~ dabc, abc, bc, c

Check if some same string is in prefix & suffix

abc

- generate a Π table : subsufix / longest prefix

Q. String a b a b c a b c a b a b a b d

pattern

1	2	3	4	5
a	b	a	b	d
0	0	1	2	0

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a	b	a	b	c	a	b	c	a	b	a	b	a	b	d
a	b	a	b	d x										
		a	b	a x										
				a x										
					a	b	a x							
							a x							
								a	b	a	b	d x		
										a	b	a	b	d

5 pe mismatch i.e. 4 tak matched. Check prefix table for 4: 4-2 = 2 steps shift se. saves comparison.

Q. | a | b | c | d | a | b | | a | b | c | d | a | b | c | d | a | b | d | e |

| a | b | c | d | a | b | d |

Prefix table.

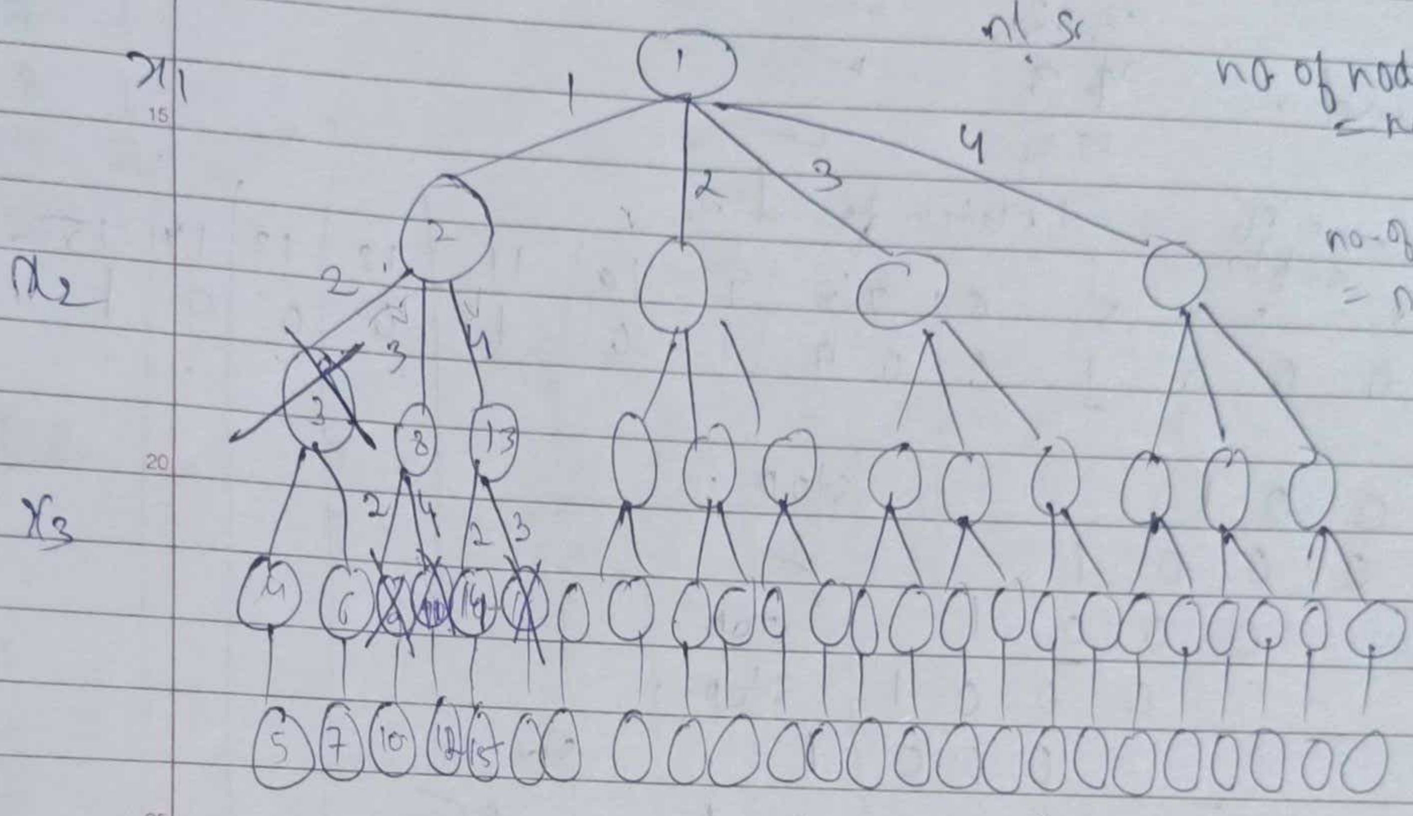
1	2	3	4	5	6	7
a	b	c	d	a	b	d
0	0	0	0	1	2	4

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
a	b	c	d	a	b		a	b	c	d	a	b	c	d	a	b	d
a	b	c	d	a	b	d×											
				a	b	c×											
						a×											
							a	b	c	d	a	b	d×				
											a	b	c	d	a	b	d

1st character has mismatch : shift once.

how many characters matched? = 2

∴ 2 : 6 : 0 ∴ 2−0 = no. of shifts to be made



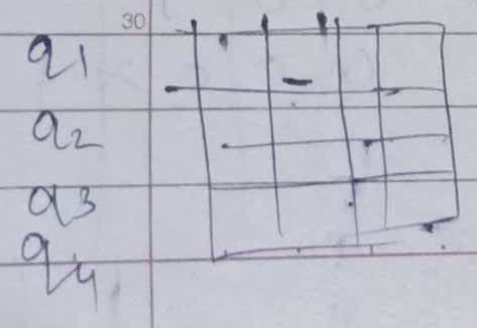
no of nodes present = no. of problem state

no. of leaf nodes = n! +

no. of nodes = $1 + \sum_{i=0}^{n-1}$

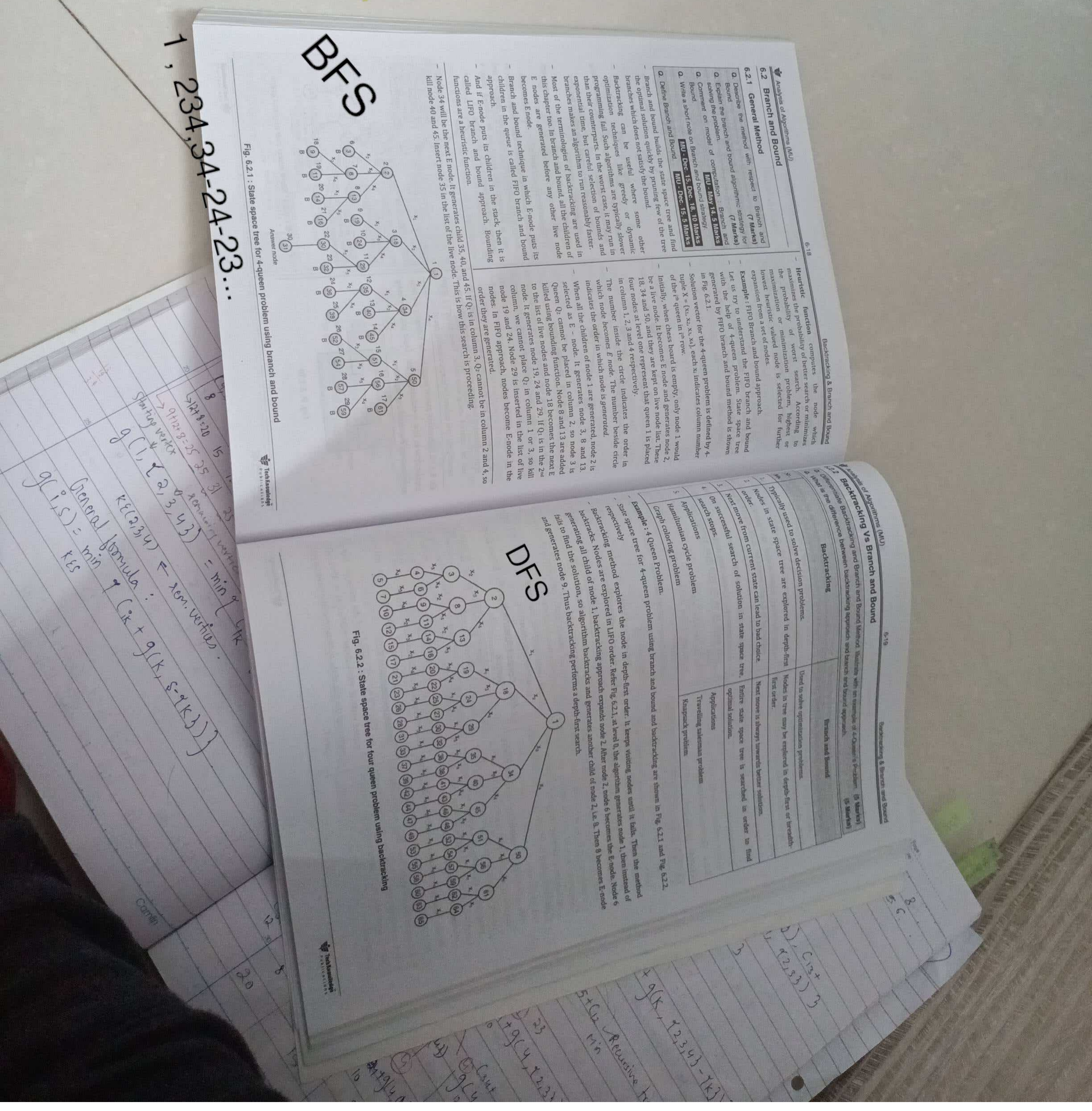
X are the nodes that get killed when applying bounding fund.

$x_1 = 2$ means putting x_1 queen1 in 2nd column.

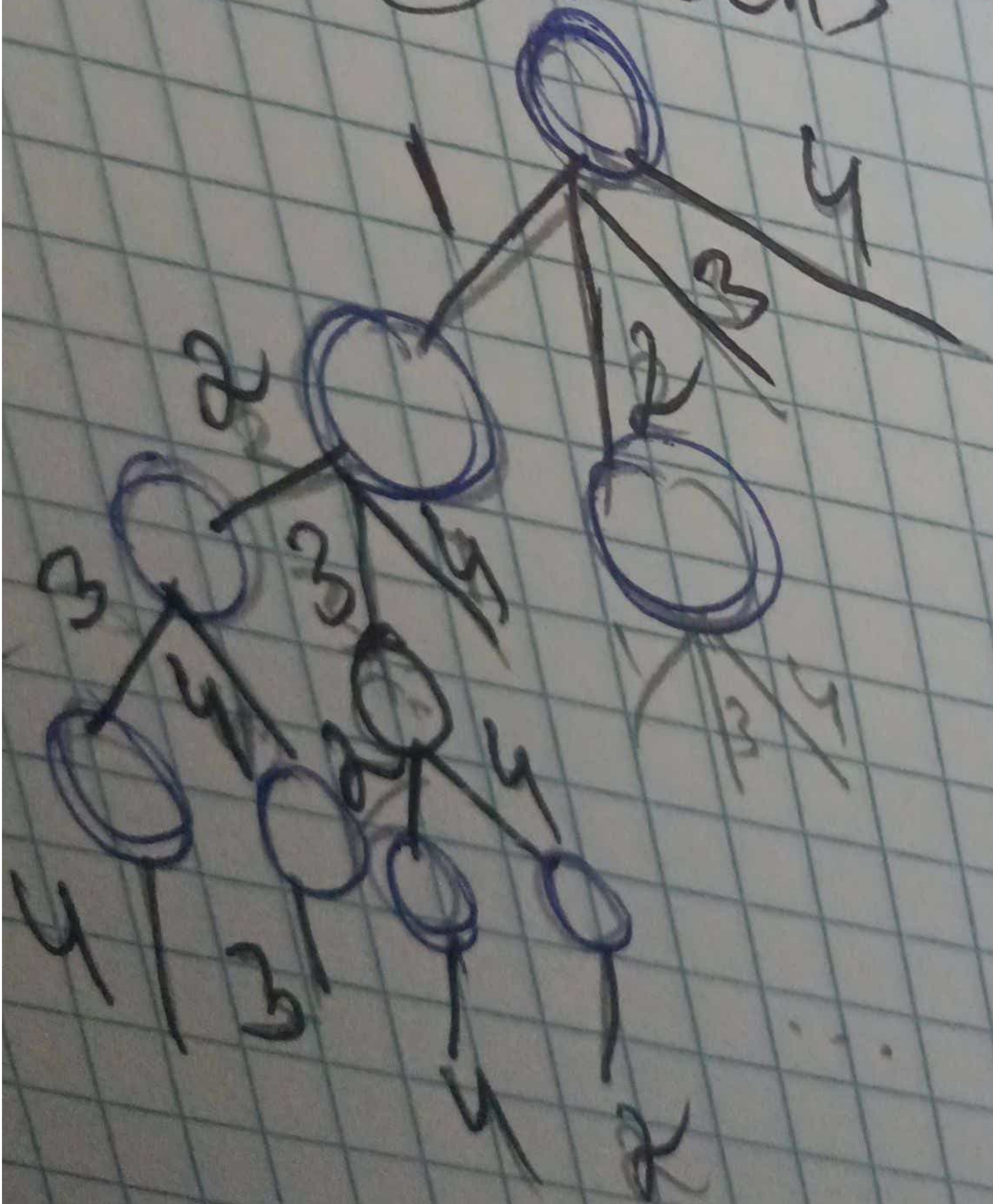


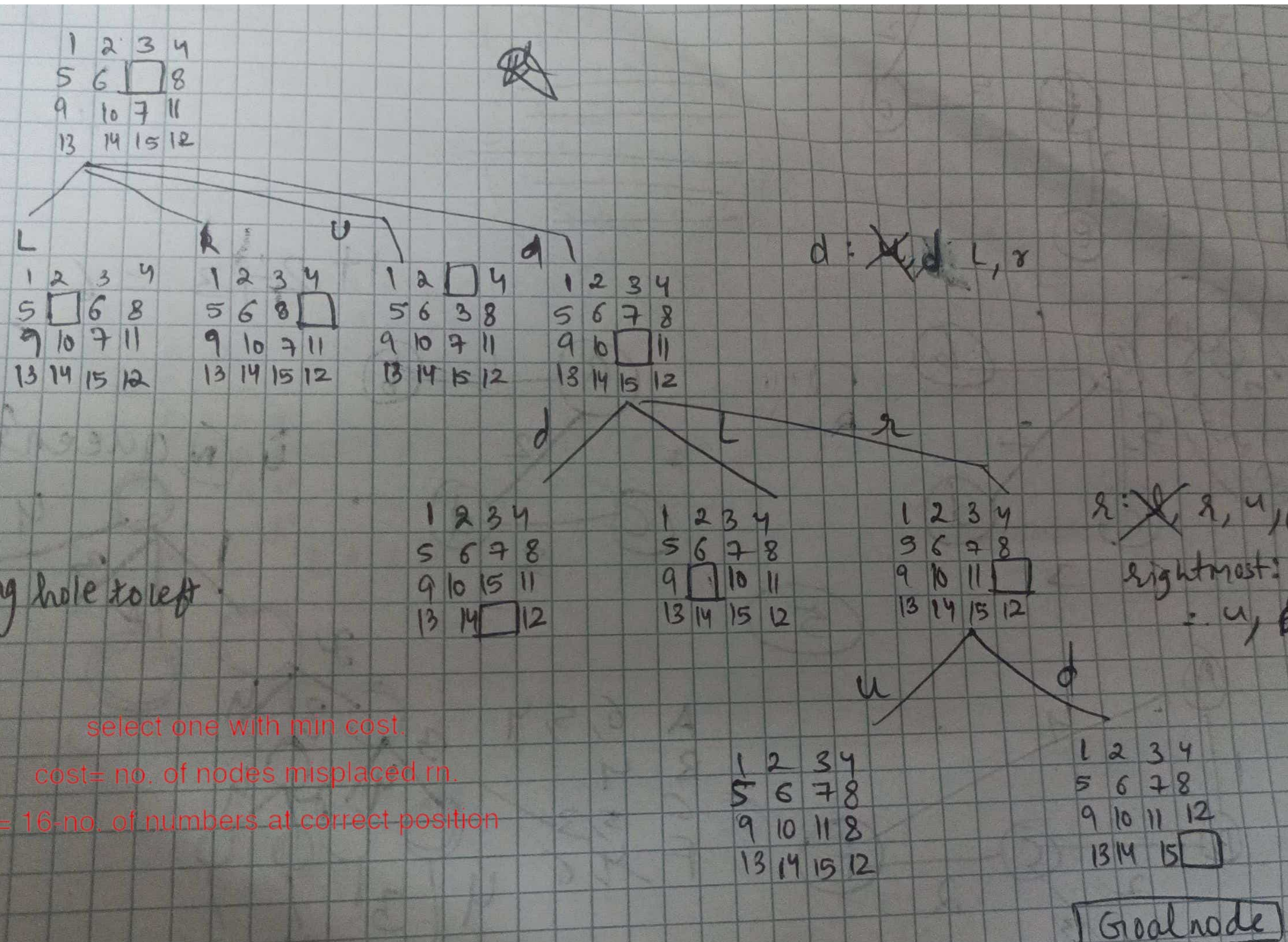
backtracking is dfs

branch and bound is bfs



4 N Queen's





d : ~~u~~, d, L, r

r : ~~L~~, r, u, d

rightmost: ~~r~~

: u, d

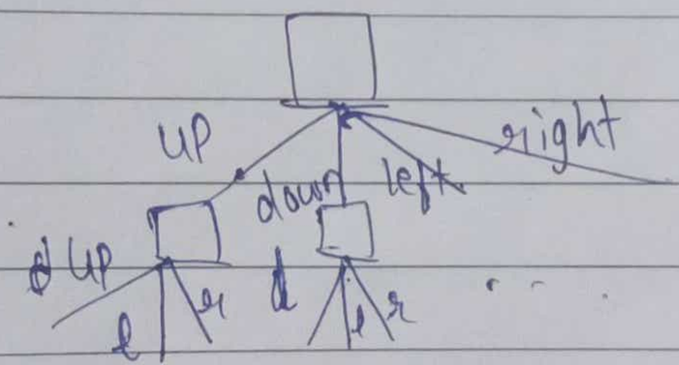
L: Bring hole to left

select one with min cost.

cost= no. of nodes misplaced rn.

= 16-no. of numbers at correct position

Branch the node with min misplace
Bound the other nodes,



up ka child down ni hoskte & viceversa
left right

a/c to Empty cell on board tree banega
like agar cell already on leftmost,
we can't have left child.

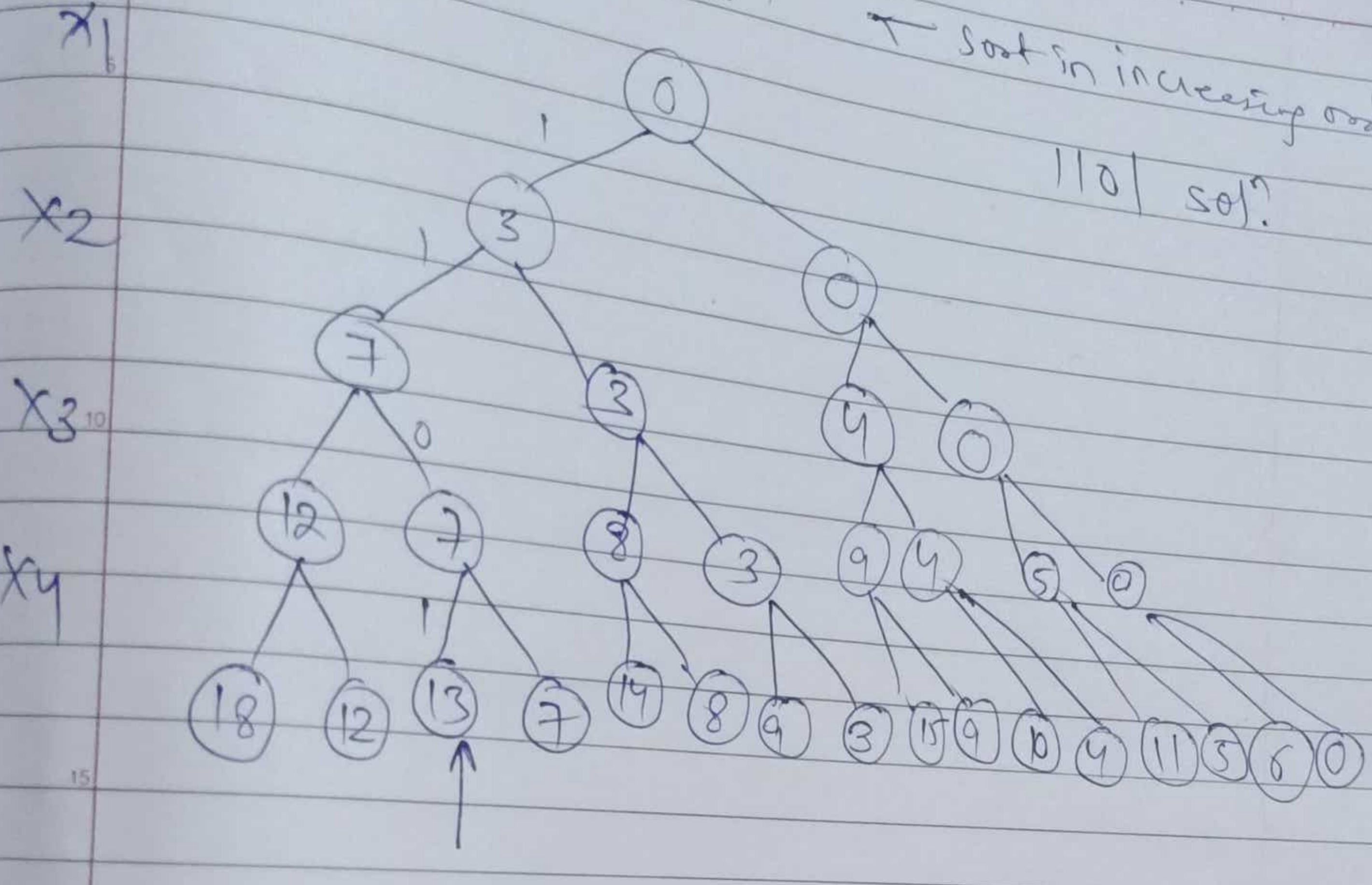
node right means space ko right side karo

$n = 4$

x_1	x_2	x_3	x_4
3	4	5	6

← Sort in increasing order

1101 solⁿ.



Leaf nodes $= 2^n$: n = no. of items in Sum of subset problem.

→ draw a state space

→ Apply Bounding funⁿ. ← queen will kill if ... } kill these.
Node > sum.

Tsp

PAGE: DATE: / /

7

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

13
25
38.

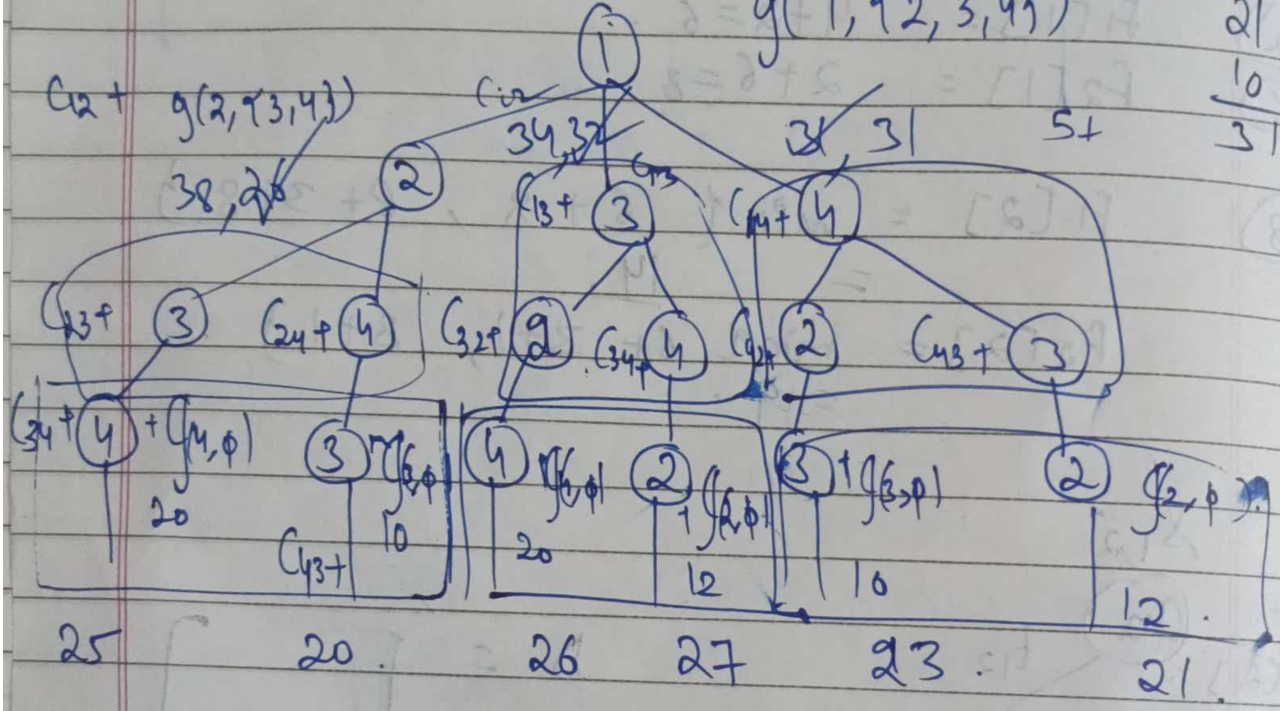
1
23
8
31

15
12
27
1
26
8
34

12
9
21
1
27.
5
32

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

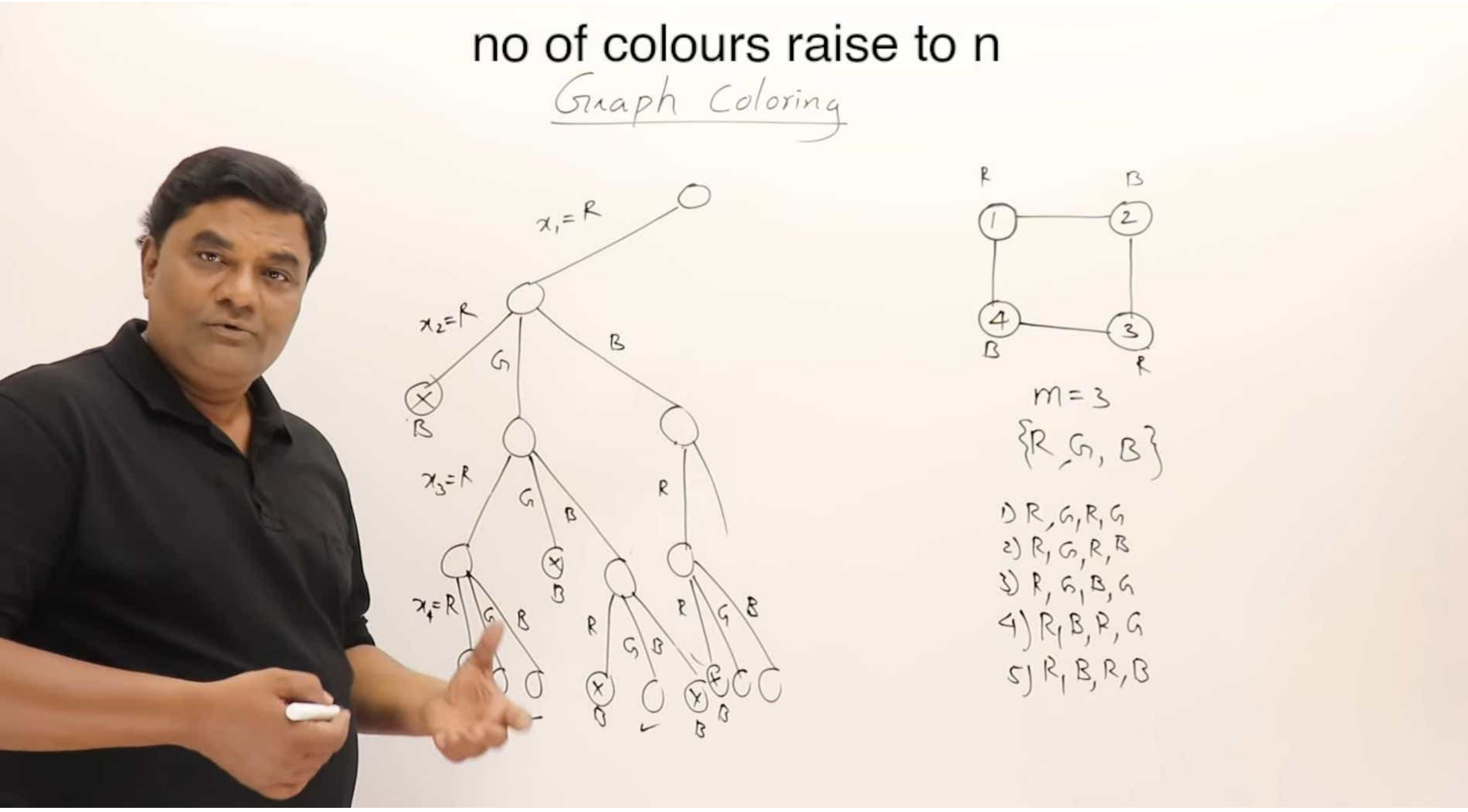
21
10
31

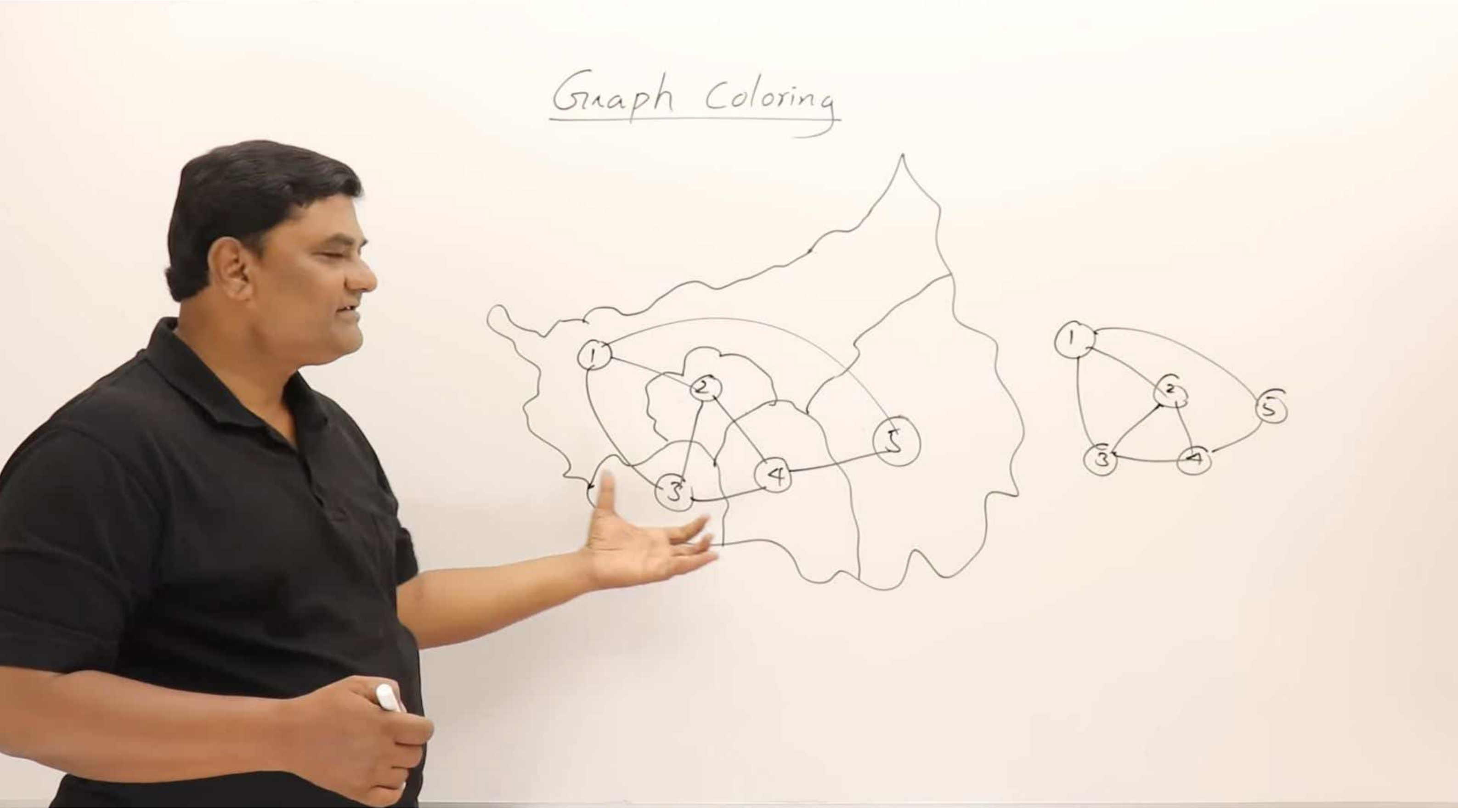


26

26
9
35

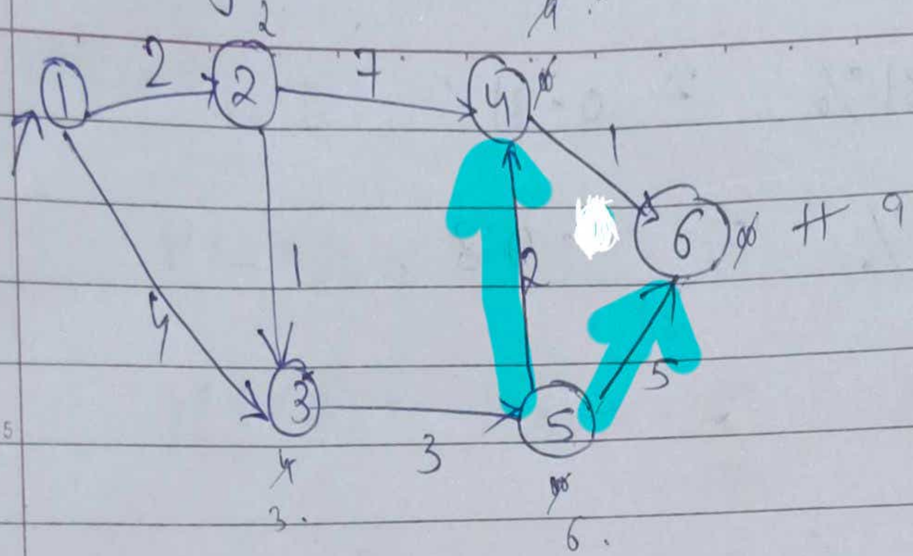
23
5
28


no of colours raise to n
Graph Coloring
x1 = R
x2 = R
G
B
x3 = R
x4 = R
m = 3
{R, G, B}
1) R, G, R, G
2) R, G, R, B
3) R, G, B, G
4) R, B, R, G
5) R, B, R, B


Graph Coloring
1
2
3
4
5
1
2
5
3
4

Dijkstra's

1\.



Sol: Source ①

source	destination					
1	2	3	4	5	6 .	

minimum select karke ..check what node is it ..

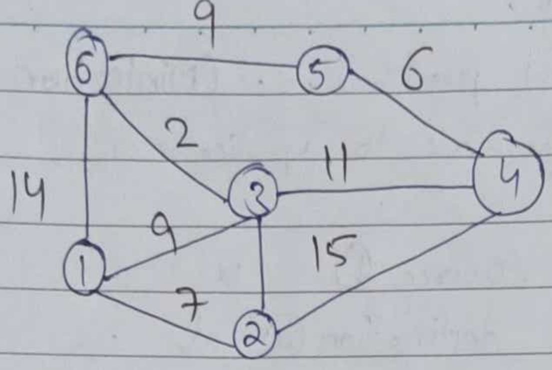
us node se kon kon connected (arrow going away) h unko relax kar

1, 2	2	~~4~~	~~∞~~	∞	∞	← min = 2.
1, 2, 3.	2	3	9	~~∞~~	∞	← min = 3.
1, 2, 3, 5.	2	3	~~9~~	6	∞ .	← min = 6.
1, 2, 3, 5, 4.	2	3	8.	6	~~11~~	← min = 8.
1, 2, 3, 5, 4, 6	2	3	8	6	9 .	

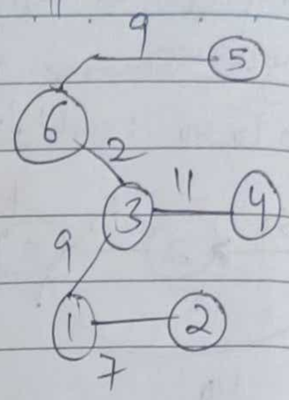
V	2	3	4	5	6
d[v]	2	3	8	6	9 .

undirected weighted :

1



o/p :



Source	destination (cost from S to D) :					
1	2	3	4	5	6	no we can go via 2
	∞	∞	∞	∞	∞	
1,	~~7~~	9	∞	∞	14=0+14	(direct wale seedha likh)
1, 2,	7	9	22	∞	14	(9 ki 2 ki b cost fix hoggi)
1, 2, 3,	7	9	20	∞	~~11~~	" 3 (minimum wala path)
1, 2, 3, 6	7	9	~~20~~	20	11 .	
1, 2, 3, 6, 4						
1, 2, 3, 6, 4, 5 .						

Source : 1, 2

1 → 3 : 1 → 3 ✓, $1 \xrightarrow{7} 2 \xrightarrow{15} 3$: x

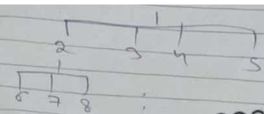
1 → 4 : $1 \underset{7}{\rightarrow} 2 \underset{15}{\rightarrow} 4$ ✓, 1 → 3 → 4 x 7+15 = 22.

Source : 1, 2, 3 .

1 → 4 : $1 \underset{7}{\rightarrow} 2 \underset{15}{\rightarrow} 4$ (22) x , $1 \underset{9}{\rightarrow} 3 \underset{11}{\rightarrow} 4$ (20) ✓

1 → 6 : $1 \underset{9}{\rightarrow} 3 \underset{2}{\rightarrow} 6$. (11) .

1 → 5 : : 1 → 6 → 5 x (23) , 1 → 3 + 6 → 9 ✓ 20



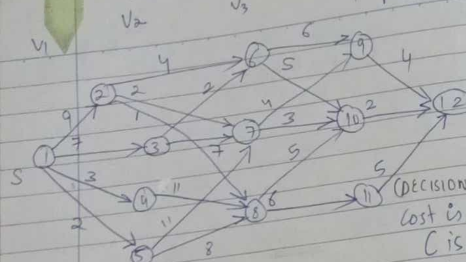
select a minimum cost node from one stage

and keep going till last node

FOR EDUCATIONAL USE

used for Resource allocation.

MULTISTAGE GRAPHS



5 stages

(Sync) (end)

(DECISIONS: FORWARD METHOD)

cost is cost of Vertex

C is cost of Edge (weight).

(useful for representing resource allocation)

→ minimiz^n. (optimiz^n problem → dynamic pgmig)

works on principal of optimality

problm must be solved in a sequence of decisions

FINDING COST & d of All vertices: BACKWARD, DATA CALC^n. no decision.

V	1	2	3	4	5	6	7	8	9	10	11	12
min Cost	16	7	9	18	15	7	5	7	4	2	5	0
vertex going min cost d	2 or 3	7	6	8	8	10	8	10	12	12	12	12
stage	V1	V2	V2	V2	V2	V3	V3	V3	V4	V4	V4	V5

cost(4,9) = 4

cost(stage, vertex)

cost(5, 12) = 0 . cost (V5,12) 12 to 12

cost (4,9) = 4 cost (V4,9) : 9 to 12.

cost (4,10) = 2 cost (V4,10) 10 to 12

cost (4,11) = 5 cost (V4,11) . 11 to 12

V3,6 → V4,9

V3,6 → V4 → 10

cost(3,6) = min{c(6,9) + cost(4,9), c(6,10) + cost(4,10)} 6 to 9 / 6 to 10

= min{6+4, 5+2} cost

= min{10, 7} 7 gives min : vertex 10.

d

10 gives Max : vertex 9.

V3,7 → V4,9

V3,7 → V4,10

cost(3,7) = min{c(7,9) + cost(4,9), c(7,10) + cost(4,10)}

= min{4+4, 3+2}

= 8. 5 cost of 7.

10 gives smaller value min . due to vertex 10

8 gives bigger value. vertex 1 d of 7

(overlapping subproblem)

using one value to find others

V3,8 → V4, 10

V3,8 → V4, 11

cost(3,8) = min{c(8,10) + cost(4,10), c(8,11) + cost(4,11)}.

= min{5+2, 6+5}

min = 7 → cost.

7 is due to (V4,10). → vertex 10.

V2,2 → V3, 6

V2,2 → V3, 7

V2, 2 → V3, 8

cost(2,2) = min{c(2,6) + cost(3,6), c(2,7) + cost(3,7), c(2,8) + cost(3,8)}

= min{4 + 7 = 11, 2 + 5 = 7, 1 + 7 = 8.}

= 7 due to 7th vertex.

V2,3 → V3, 6

V2,3 → V3, 7

cost(2,3) = min{c(3,6) + cost(3,6), c(3,7) + cost(3,7)}

= min{2+7, 7+5}

= 9 due to vertex 6

V2,4 → V3, 8 | cost(2,4) = min{c(4,8) + cost(3,8)}

= 11 + 7 = 18 . due to vertex 8

V2,5 → V3,7

V2,5 → V3,8 .

cost(2,5) = min{c(5,7) + cost(3,7), c(5,8) + cost(3,8)}. 15 due to 8th vertex

= 11+5, 8+7

Camlin

Dynamic Programming

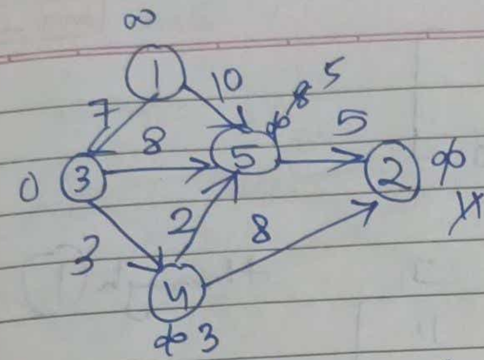
3. If entire item cannot be | If item cannot be fully accommodated then

no. of iterations = |V|-1

PAGE:
DATE: / /

(2,3)
(3,5)

Q. SSSP (BF).



MCP from 3 to 2 ?

~~11~~ 10.

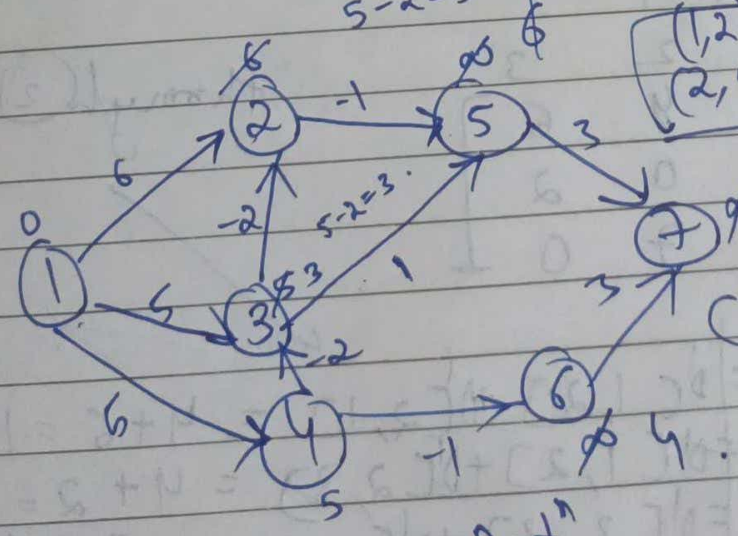
(1,3) (1,5) ~~t2~~ (3,4) (3,5) (4,2) (4,5) (5,2).

Source	destn:			
3	1	2	4	5
	∞	∞	∞	∞
→	∞	∞	3	8
	∞	11	3	5
	∞	10	3	5

→ check for negative edge cycle

5-2=3.

Q.



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

Source 1 Shortest path = ?

(5,7) (6,7) 9.

Source	Destn.						
1	1	2	3	4	5	6	7
	∞	∞	∞	∞	∞	∞	∞
	0	∞	∞	∞	∞	∞	∞
	0	6	5	5	∞	∞	9
	0	3	3	5	6	4	7
	0	3	3	5	6	4	7

0 3 3 5 2 4 5

floyd warshall

PAGE: DATE: / /

$$D^0 = \begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & \infty & 0 \end{bmatrix} \end{array}$$

D1 ke liye row1 and column 1 as it is

through 1 .

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

diagonals will be 0 since no loops

min out of (2,17)

$D^0[2,3] = 2$

$D^0[2,1] + D^0[1,3]$ via 1 in D1

$= 6 + 11$

$= 17.$

baki elements we have to find by selecting

min of D matrix and Dn matrix via n

$D^0[3,2] = \infty$.

$D^0[3,1] + D^0[1,2]$

$= 3 + 4$

$= 7.$

through 2 .

$$D^2 = \begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0 & 4 & 6 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix} \end{array}$$

D1 is called iteration 1

D2 iteration 2

0 $D^1[1,1] = D^1[1,2] + D^1[2,1] = 4 + 6 = 10.$

$\frac{1}{3}$ $D^1[1,3] = D^1[1,2] + D^1[2,3] = 4 + 2 = 6$ ✓

$D^1[3,1] = D^1[3,2] + D^1[2,1] = 7 + 6 = 13$

0 $D^1[3,3] = D^1[3,2] + D^1[2,3] = 7 + 2 = 9.$

find D3.

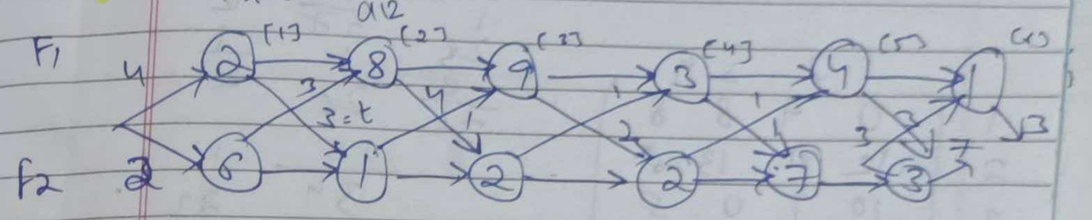
ALS

ALS . (a_{ij})

PAGE: DATE: / /

$\cancel{D^8} =$

$4+2=6$. 14



$2+6=8$

(2) $f_1[1] = 4+2 = 6.$

(6) $f_2[1] = 2+6 = 8.$

(8) $f_1[2] = \min\{6+8, 8+3+8\}$

$= 14.$

$f_2[2] = \min\{6+3+1, 8+1\}.$

$= 9.$

$l_{1,2}$

$f_1[2]$. a_{12} t_{12}

$D = \begin{bmatrix} & \\ & \end{bmatrix}$

$= D^0$

if no. of vertices = 4 : find till D^4

$D^1 \to$ via 1 | $D^0[\,.\,]$ ...

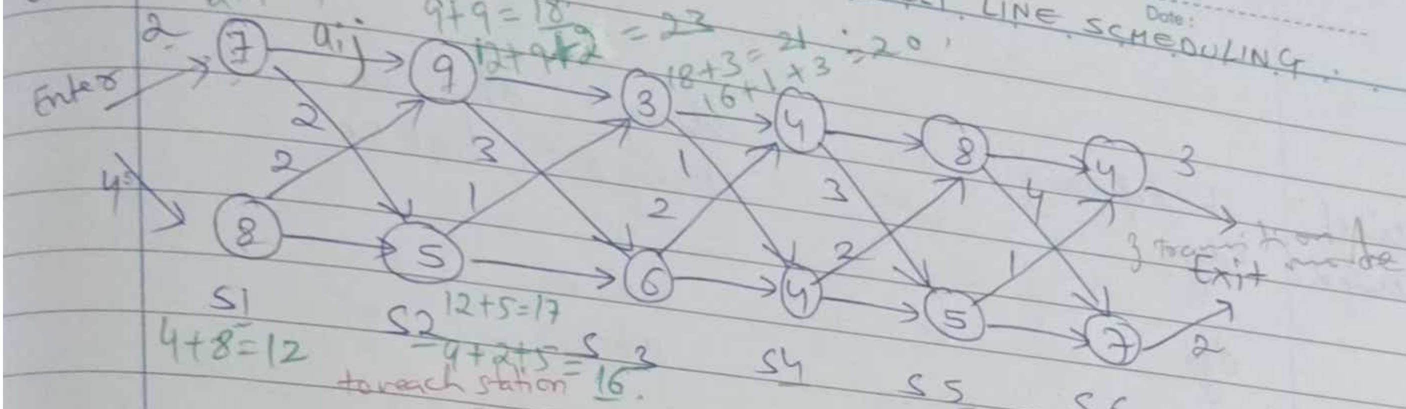
maintain line selected table and min cost table

(aij)

Minimiz. prblm

ASSEMBELY LINE SCHEDULING.

Q.1.



		7	9	3	4	8	4	
		S1	S2	S3	S4	S5	S6	
Station 1	$f_1[j]$	9	18	20	24	32	35	(upper)
Station 2	$f_2[j]$	12	16	22	25	30	37	(lower).
		8	5	6	4	5	7	

		1	2	3	4	5	6
Line 1	$l_1[j]$	1	1	2	1	1	2
Line 2	$l_2[j]$	2	1	2	1	2	2

$n = 6$

$f^* = \min\{35+3, 37+2\}$

$f^* = 38.$

Page
Date

- Using dynamic pgrmng

→ bottom up approach but table will be filled from top-down

if $(A[i] = B[j])$
  $LCS[i,j] = 1 + LCS[i-1, j-1]$
else
  $LCS[i,j] = \max(LCS[i-1,j], LCS[i,j-1])$

Q.1. A = | b | d | (1, 2)

B = | a | b | c | d | (1, 2, 3, 4)

→

		0	1 (a)	2 (b)	3 (c)	4 (d)
	0	0	0	0	0	0
b	1	0	0 (b,a)	0+1=1 (b,b)	1 (c,b)	1 (b,d)
d	2	0	0 (a,d)	1 (d,b)	1 (c,d)	1+1=2 (d,d)

b d

Equal pe jao (sirf diagonal wale likh.)

∴ Size of the Longest common subsequence is 2

Traceback to find what the longest common subsequence is

total no. of time take = $O(m \times n)$.

match : left diag + 1

Q.2. Str1 = stone mismatch : max(i-1,j-1)

Str2 = longest

Y on x axis

X on y axis

		0	1 (l)	2 (o)	3 (n)	4 (g)	5 (e)	6 (s)	7 (t)
	0	0	0	0	0	0	0	0	0
s	1	0	0	0	0	0	0	1	1
t	2	0	0	0	0	0	0	1	2
o	3	0	0	1	1	1	1	1	2
n	4	0	0	1	2	2	2	2	2
e	5	0	0	1	2	2	3	3	3

o n e

Size of table = $m \times n$
time taken = $O(mn)$.

LCS

X = ACBAED , Y = ABCABE

X ↓

Y →

		0	1	2	3	4	5	6
		0	A	B	C	A	B	E
0	0	0	0	0	0	0	0	0
1	A	0	1	1	1	1	1	1
2	C	0	1	1	2	2	2	2
3	B	0	1	2	2	2	3	3
4	A	0	1	2	2	3	3	3
5	E	0	1	2	2	3	3	4
6	D	0	1	2	2	3	3	4

first time ever 1,2,3,4 kaha aya circle them..

uske corresponding alphabet likh neeche

A B C A B E

A C B E

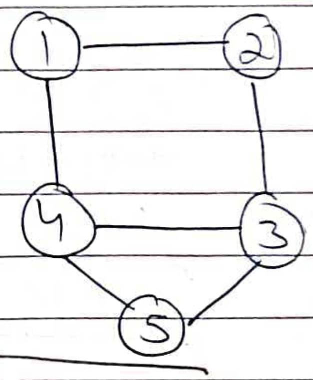
TSP

LC BB

Branch & Bound

Cost Adjacency Matrix : $\begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix}$

Graph :



1. REDUE ROW

$\begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix}$ Min row value = 10, 2, 2, 3, 4

Subtract Min row value from Resp. Row.

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

1 0 3 0 0

Total cost of Reduction of all ROWS = 10+2+2+3+4 = 21.

2. REDUE COLUMNS.

total cost of reduction = row cost reduction+column cost reduction

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

cost of Red^n. of all col^n = 25.

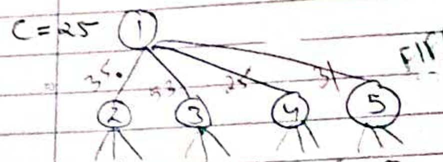
11 aya n column1 cost toh subtract 11 from it

visited to ② means route from 1 to 2.

Hence the above Matrix is a Reduced matrix
i.e. it is having atleast one Element as 0.

What we did is
→ we found shortest distances for Matrices & subtracted it from matrix. ∴ Min cost of tour will be 25.

C = 25



upper bound = ∞.

FIFO

1→2, 1→3, 1→4, 1→5

$\boxed{2 \to 3,4,5 \quad 3 \to 2,4,5 \ldots}$

cost of 2nd node i.e. ①—② will be found by making 1st row & 2nd column values as ∞.
(we will have 1→2 & 2→1 as ∞ too)

①—②

$$
\begin{array}{c|ccccc|c}
 & 1 & 2 & 3 & 4 & 5 & \\
1 & \infty & \infty & \infty & \infty & \infty & \infty \\
2 & \infty & \infty & 11 & 2 & 0 & 0 \\
3 & 0 & \infty & \infty & 0 & 2 & 0 \\
4 & 15 & \infty & 12 & \infty & 0 & 0 \\
5 & 11 & \infty & 0 & 12 & \infty & 0 \\
 & 0 & \infty & 0 & 0 & 0 &
\end{array}
$$

$C(1,2) + r + \hat{r}$
$= 10 + 25 + \underline{0}$.
$r = C(1)$.

35

①—③

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

Make 3→1 as ∞ too.

⇓

Reduce this matrix.

$$
\begin{bmatrix}
\infty & \infty & \infty & \infty & \infty \\
1 & \infty & \infty & 2 & 0 \\
\infty & 3 & \infty & 0 & 2 \\
4 & 3 & \infty & \infty & 0 \\
0 & 0 & \infty & 12 & 0
\end{bmatrix}
$$

$C(1,3) + r + \hat{r}$
$= 17 + 25 + 11$
$= \underline{53}$.

①—④

$$
\begin{array}{c|ccccc|c}
 & 1 & 2 & 3 & 4 & 5 & \\
1 & \infty & \infty & \infty & \infty & \infty & 0 \\
2 & 12 & \infty & 11 & \infty & 0 & 0 \\
3 & 0 & 3 & \infty & \infty & 2 & 0 \\
4 & 15 & 3 & 12 & \infty & 0 & 0 \\
5 & 11 & 0 & 0 & \infty & \infty & 0 \\
 & 0 & 0 & 0 & 0 & 0 &
\end{array}
$$

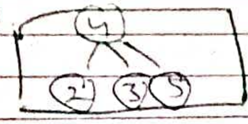
$C(1,4) + r + \hat{r}$
$= 0 + 25 + 0$.
$= \underline{25}$.

①—⑤

$$
\begin{array}{c|ccccc|c}
 & 1 & 2 & 3 & 4 & 5 & \\
1 & \infty & \infty & \infty & \infty & \infty & 0 \\
2 & 12 & \infty & 11 & 2 & \infty & 2 \\
3 & 0 & 3 & \infty & 0 & \infty & 0 \\
4 & 15 & 3 & 12 & \infty & \infty & 3 \\
5 & 11 & 0 & 0 & 12 & \infty & 0 \\
 & 0 & 0 & 0 & 0 & 0 &
\end{array}
\rightarrow
\begin{array}{c|ccccc|c}
 & 1 & 2 & 3 & 4 & 5 & \\
1 & \infty & \infty & \infty & \infty & \infty & 0 \\
2 & 10 & \infty & 9 & 0 & \infty & 0 \\
3 & 0 & 3 & \infty & 0 & \infty & 0 \\
4 & 12 & 0 & 9 & \infty & \infty & 0 \\
5 & 11 & 0 & 0 & 12 & \infty & 0 \\
 & 0 & 0 & 0 & 0 & 0 &
\end{array}
$$

$C(1,5) + r + \hat{r} = 1 + 25 + 5$
$= 31$.

Min of all cost from 1 to 2,3,4,5 = 25 of node 4



LC

use ①—④ matrix.

Select edge 4-3 (Path 1-4-3)

Set $M_4[1][\,] = M_4[4][\,] = M_4[\,][3] = \infty$

Set $M[3][1] = \infty$

Reduce the resultant matrix if required. row red^n. cost = 2

$$M_4 \Rightarrow \begin{array}{|c|c|c|c|c|l}
\hline
\infty & \infty & \infty & \infty & \infty & \rightarrow x \\
\hline
12 & \infty & \infty & \infty & 0 & \rightarrow 0 \\
\hline
\infty & 3 & \infty & \infty & 2 & \rightarrow \boxed{2} \\
\hline
\infty & \infty & \infty & \infty & \infty & \rightarrow \infty \\
\hline
11 & 0 & \infty & \infty & \infty & \rightarrow 0 \\
\hline
\end{array}
\Rightarrow
\begin{array}{|c|c|c|c|c|}
\hline
\infty & \infty & \infty & \infty & \infty \\
\hline
12 & \infty & \infty & \infty & 0 \\
\hline
\infty & 1 & \infty & \infty & 0 \\
\hline
\infty & \infty & \infty & \infty & \infty \\
\hline
11 & 0 & \infty & \infty & \infty \\
\hline
\multicolumn{1}{c}{\downarrow} & \multicolumn{1}{c}{\downarrow} & \multicolumn{1}{c}{\downarrow} & \multicolumn{1}{c}{\downarrow} & \multicolumn{1}{c}{\downarrow} \\
\multicolumn{1}{c}{11} & \multicolumn{1}{c}{0} & \multicolumn{1}{c}{x} & \multicolumn{1}{c}{x} & \multicolumn{1}{c}{0}
\end{array}
= M_7'$$

col red^n. cost =11

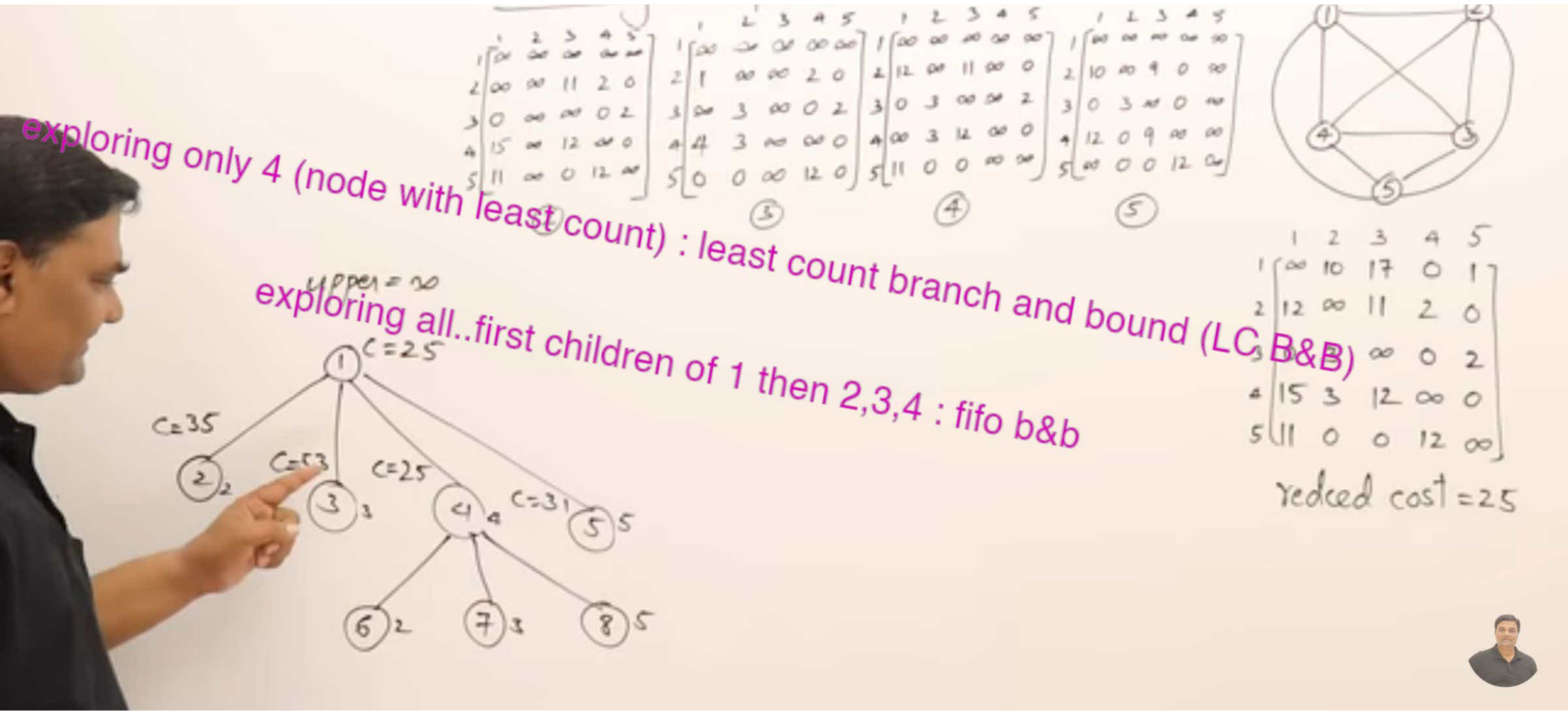
M_7' is not reduced. Reduce it by subtracting 11 from column 1.

$$\therefore M_7' \Rightarrow \begin{array}{|c|c|c|c|c|}
\hline
\infty & \infty & \infty & \infty & \infty \\
\hline
1 & \infty & \infty & \infty & 0 \\
\hline
\infty & 1 & \infty & \infty & 2 \\
\hline
\infty & \infty & \infty & \infty & \infty \\
\hline
0 & 0 & \infty & \infty & \infty \\
\hline
\end{array} = M_7$$

Cost of node 7

$C(7) = C(4) + \text{Reduction cost} + M_4[4][3]$

$= 25 + \boxed{2} + 11 + 12 = 50$

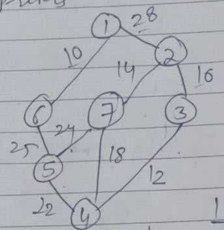

exploring only 4 (node with least count) : least count branch and bound (LC B&B)
exploring all..first children of 1 then 2,3,4 : fifo b&b
upper = ∞
c=25
c=35
c=53
c=25
c=31
redced cost=25

GRAPH MUST BE CONNECTED OR ELSE WE CAN'T FIND MCST.

Greedy way to find min cost spanning tree → ① Prims
② Kruskal's.

① Prims.

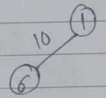
7 vertices
9 Edges



→ select a min. cost edge.
→ select a min cost edge & make sure it is connected to already selected vertices.

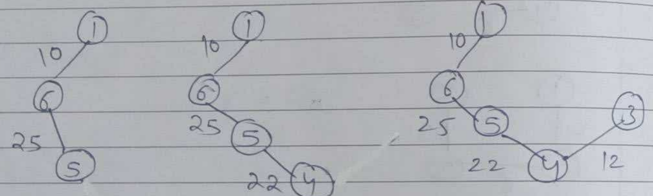
cost of Edges:
10, 28, 16, 12, 22, 25, 18, 14, 24
Torder: ~~10~~, ~~12~~, ~~14~~, ~~16~~, ~~18~~, ~~22~~, 24, ~~25~~, 28

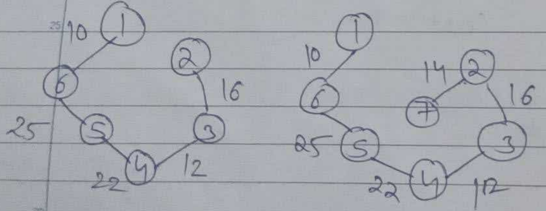
Step 1:



if now we select 12, ④ & ③ aren't connected to ① or ⑥

∴



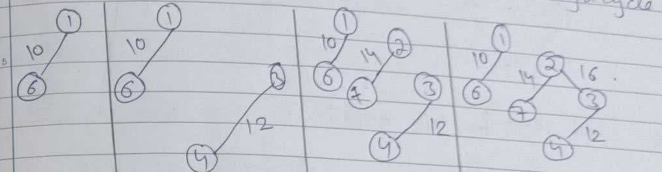


7 Vertices.
6 Edges.

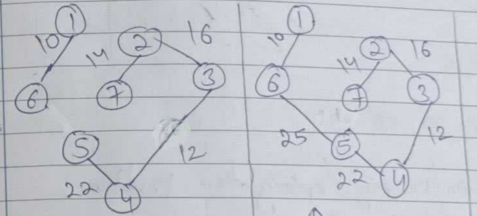
↑ MCST cost = 99.

~~10~~, ~~12~~, ~~14~~, ~~16~~, 18, ~~22~~, 24, ~~25~~, 28

② Kruskal's
→ always select a min. cost edge without forming a cycle.



Now considering edge 18 will give us a cycle ∴ discard 18



↑
MCST cost = 99.
(Same result as prim's).

$\Theta(|V||E|) = \Theta(n \cdot e) = \Theta(n^2)$ → Kruskal's.
But min Heap (gives always a min value) can improve Kruskal's complexity. If all these edges are kept in a MinHeap, deleting one will give us the next min (w/o having to search for it) in $O(\log n)$. ∴ we have to do it n time

∴ $O(n \log n)$ → Kruskal's Reduced Time comp. using MinHeap

Kruskal may work for nonconnected graph but it will give o/p for each component separately.

no. of edges in mcst = no. of vertices in original graph - 1

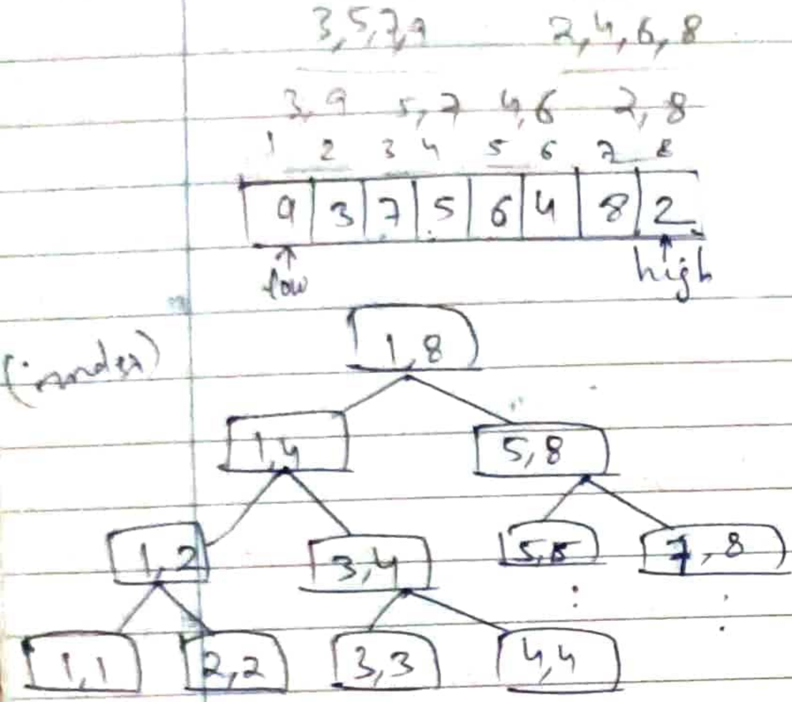
40	20	60	10	50	30

$n = 6$.

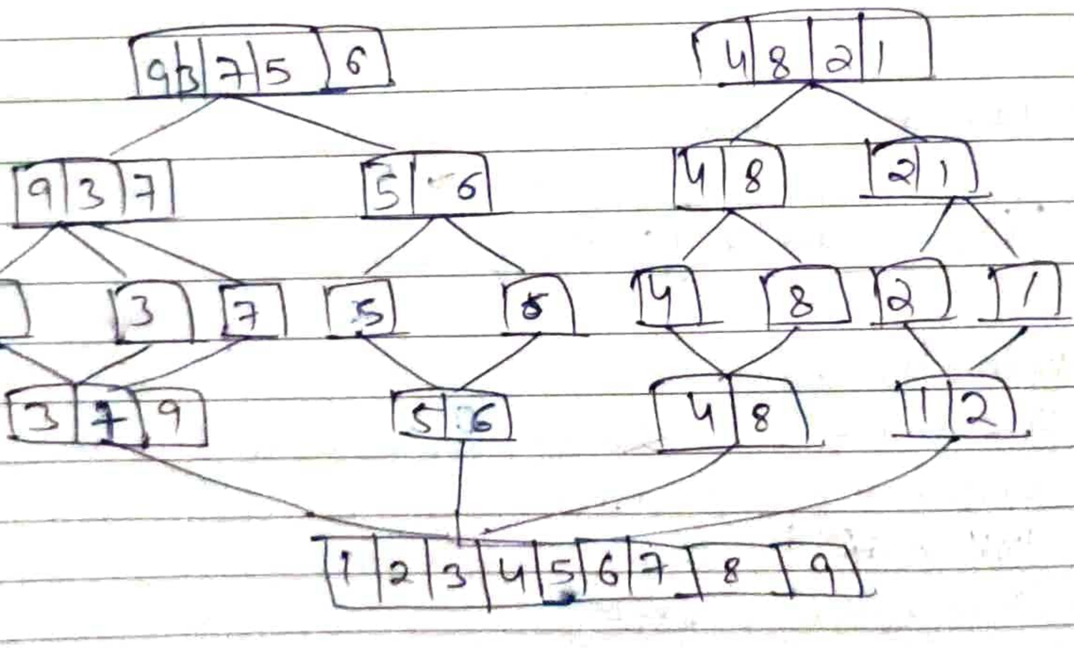
1. 20, 40, 60, 10, 50, 30.
2. 20, 40, 10, 60, 50, 30.
   20, 10, 40, 60, 50, 30.
   10, 20, 40, 60, 50, 30.
3. 10, 20, 40, 50, 60, 30.
4. 10, 20, 40, 50, 30, 60.
   10, 20, 40, 30, 50, 60.
   10, 20, 30, 40, 50, 60.

Best case : $O(n)$ comp., $O(1)$ swaps.
Worst case : $O(n^2)$ comp. & swaps
Avg case : $O(n^2)$ comp & swaps

MERGE SORT



Algorithm mergesort(l, h)
{ if (l<h)
{ mid = (l+h)/2 ;
mergesort(l, mid);
mergesort(mid+1, h);
merge (l, mid, h);
}
}



- Example of Insertion sort :

20, 100, 3, 25, 6, 95, 45, 55 .

→ 3, 20, 100, 25, 6, 95, 45, 55 → 3, 20, 25, 100, 6, 95, 45, 55

→ 3, 6, 20, 25, 100, 95, 45, 55 → 3, 6, 20, 25, 95, 100, 45, 55 .

→ 3, 6, 20, 25, 45, 95, 100, 55 → 3, 6, 20, 25, 45, 55, 95, 100.

- Example of selection sort :

20, 100, 3, 25, 6, 95, 45, 55 . $n = 8$.

min next min. lowest swap with 0th .

→ 3, 100, 20, 25, 6, 95, 45, 55

→ 3, 6, 20, 25, 100, 95, 45, 55

→ 3, 6, 20, 25, 45, 95, 100, 55

→ 3, 6, 20, 25, 45, 55, 100, 95

→ 3, 6, 20, 25, 45, 55, 95, 100 .

Quicksort Example:

44, 88, 77, 22, 66, 11, 99, 55, 00, 33

1. 44 = pivot . < pivot | pivot | > pivot

44 < 88 (j++), 44 < 77 (j++), 44 > 22 . i = 3 . Find element less than pivot

++i . i = 1 . Swap i & j elements .

44, 22, 77, 88, 66, 11, 99, 55, 00, 33 .

j = 3 j = 5

proceed similarly, j++ until u find Elem. < pivot .

j = 5 pe 11 . i++ ⇒ i = 2 . swap .

44, 22, 11, 88, 66, 77, 99, 55, 00, 33 .

j = 5 j = 7 j = 8

i = 3 & j = 8 swap

44, 22, 11, 00, 66, 77, 99, 55, 88, 33 .

j = 9 .

j = 9, i = 4 .

~~44, 22, 11, 00, 66, 77, 99, 88, 88, 33~~ .

~~j = 9, i = 5~~

~~44, 22, 11, 55, 00, 33, 99, 88, 66, 77~~ .

i = 4, j = 9

44, 22, 11, 00, 33, 77, 99, 88, 66 .

swap p & i .

33, 22, 11, 00, 44, 77, 99, 88, 66 .

0 1 2 3 0 1 2 3 .

pivot = 33

22 < 33 ; i++ ⇒ i = 1 .

| 33 | 22 | 11 | 00 |

i = 1, j = 1 swap . : No change .

j++ 11 < 33 . i++ ⇒ i = 2, j = 2 . swap

j++, 00 < 33 i++ ⇒ i = 3, j = 3 . swap .

Now swap p & i . ∵ we reached end of string : i = n-1

| 00 | 22 | 11 | 33 |

p i j

all Elements > P ∴ p++

| 00 | 22 | 11 | 33 |

11 < 22 i = 2, j = 2 .

Swap i & p .

| 00 | 11 | 22 | 33 |

$$\sum_{i=1}^{n} x_i w_i \le m.$$

F. Knapsack — Greedy

Page :
Date :

Q.2.

objects	obj 1	2	3	4	5	6	7
p_i	10	5	15	7	6	18	3
w_i	2	3	5	7	1	4	1
p_i/w_i	5	1.6	3	1	6	4.5	3
x_i	1	$\frac{2}{3}$ rd	1	0	1	1	1

M = 15.

increasing order of p_i/w_i :

I4 I2 I3 I7 I6 I1 I5

1, 1.6, 3, 3, 4.5, 5, 6.

portion	knapsack items	profit	weight left =
	ϕ	0	15
1	include I5	6	15 − 1 = 14
1	I5, I1	10+6=16	14−2 = 12
1	I5, I1, I6	16+18=34	12−4 = 8
1	I5, I1, I6, I7	34+3 = 37	8−1 = 7
1	I3	37+15 = 52	7−5 = 2
$\frac{2}{3} \times I_2$	I2	$52 + \frac{2}{3} \times 5 = 55.33$	$2 - \frac{2}{3} \times 3 = 0$
0	I4		

weight left in knaps = 2.

weight of I2 = 3.

$\therefore \frac{2}{3}$ rd of I_2

x_i = portion.

$$\text{Weight} = \sum_{i=1}^{n} x_i \cdot w_i = 1\times1 + 1\times2 + \frac{2}{3}\times3 + 5\times1 + 7\times0$$

$$+ 4\times1 + 1\times1 = \underline{\underline{15}}. = M$$

$$\text{profit} = \sum_{i=1}^{n} x_i \cdot p_i = 1\times10 + \frac{2}{3}\times5 + 1\times15 + 0\times7 + 1\times6 +$$

$$1\times18 + 1\times3 = \underline{55.33 \text{ units}}$$

Page :
Date :

(2) Binary 1/0 . x_i can be 0 or 1 .

$p = \{1, 2, 5, 6\}$ $n = 4$, $m = 8$.
$w = \{2, 3, 4, 5\}$.

for n objects, there are 2^n combinations/options, $O(2^n)$.

TABULAR METHOD

profits

p_i	w_i		V	0	1	2	3	4	5	6	7	8
		0	0	0	0	0	0	0	0	0	0	0
1	2	I1	1	0	0	1	1	1	1	1	1	1
2	3	I2	2	0	0	1	2	2	3	3	3	3
5	4	I3	3	0	0	1	2	5	5	6	7	7
6	5	I4	4	0	0	1	2	5	6	6	7	8

Max profit

$\therefore$ ~~I4~~ profit left $= 8 - P_4 = 8 - 6 = 2$ some, first time 2 is in I_2 $\therefore$ ~~I2~~

- while considering 2nd object, 1st object will be considered too .
- if weight of an object is 2, it can be filled with bag with capacity 2 or more . (column 2).

for last row,

$$V[i,w] = \max\{V[i-1, w], V[i-1, w - w[i]] + P[i]\}$$

profit of that obj

N.P.

$$V[4,1] = \max\{V[3,1], V[3, 1-5] + 6\}$$
$$= \max\{0,$$

0

$$V[4,5] = \max\{V[3,5], V[3, 5-5] + p[4]\}.$$
$$= \max[5, 0+6]$$

greater.

$$V[4,6] = \max[V[3,6], V[3, 6-5] + 6]$$
$$= 6, 6 \rightarrow 6$$

$$V[4,7] = \max\{V[3,7], V[3, 7-5] + 6\}$$
$$= 7, 1+6 \rightarrow 7$$

$$V[4,8] = \max\{V[3,8], V[3, 8-5] + 6\}$$
$$= 7, 2+6$$

Camlin

job sequencing with Deadlines. — Greedy

$n = 5$

jobs	J1	J2	J3	J4	J5
profits	20	15	10	5	1
deadlines	2	2	1	3	3
	[1,2]	[0,1]	:	[2,3]	.

0 J_2 1 J_1 2 J_4 3

→ Max profit order : 20, 15, 10, 15, 1 .
decreasing

J1 can wait for 2 units
∴ put on left of 2 .

J2 can wait for 2 .
search for place on left of 2 .

J3 can wait for 1 .
but left of 1 already occupied .

J4 can wait for 3
one slot free on left of 3 .

all slots occupied .

profit = $P_1 + P_2 + P_4$
$= 20 + 15 + 5 = \underline{40}$ unit

T = PLANINGANDANALYSIS

P = AND ; j
1 2 3

i=1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
P	L	A	N	I	N	G	A	N	D	A	N	A	L	Y	S	I	S

Shift 0
A
A
AND
A
A
A
A
AND
A
A
AND
A
AN
A
A
A.

Match found at i = 8 till 10 & j = 1.

q = 13 , T = 2359023141526739921.
p = 31415.

q → random prime no.
31415 Mod 13 = 7

A % B = A Mod B
= A − A ÷ B × B
Mode : 4 : Dec

len(P) = 5

len(5) T = 23590 Mod 13 = 8
35902 Mod 13 = 9
59023 Mod 13 = 3
90231
02314
:
:

P Mod q = hash

if Hash match

if all characters of P&T match → valid hit

if all char. of P&T dont match → Spurious hit

T= 2 3 5 9 0 2 3 1 4 1 5 2 6 7 3 9 9 2 1

$(23590)(\ )\times 10 - \cancel{10}\,10^5 \times 2 + 2$.

	hash values	T mod 13.
= 2 3 5 9 0	8	
3 5 9 0 2	9	
5 9 0 2 3	3	
9 0 2 3 1	11	
0 2 3 1 4	0	
2 3 1 4 1	1	
3 1 4 1 5	7	✓ valid hit
1 4 1 5 2	8	
4 1 5 2 6	4	
1 5 2 6 7	5	
5 2 6 7 3	10	
2 6 7 3 9	11	
6 7 3 9 9	7	✓ spurious hit.
7 3 9 9 2	9	
3 9 9 2 1	11	

hash = p mod q = 31415 mod 13 = 7.

T(s+1) = 2
1st char.

$23590 \times 10 - 10^5 \times 2 + 2 = 35902$.
$35902 \times 10 - 10^5 \times 3 + 3 = 59023$.
$59023 \times 10 - 10^5 \times 5 + 1 = 90231$.
$90231 \times 10 - 10^5 \times 9 + 4 = 2314$.

index 5+5+2.
5+2
5+2.

⋮

Q. $p = \boxed{26}$, $q = 11$

no. of digits in p = 2 ∴ 2 characters consider kar in T

p mod q = 4

T= 3 1 4 1 5 9 2 6 5 3 5 8 9 7 9 3

mod 11	hash values.	
→ 3 1	9	
1 4	3	
4 1	8	
1 5	4	✓ Spurious Hit
5 9	4	✓ spurious Hit
9 2	4	✓ Spurious Hit
2 6	4	✓ valid Hit
6 5	10	
5 3	9	
3 5	2	
5 8	3	
8 9	1	
9 7	9	
7 9	2	
9 3	5	

$$LCS(i,j) = \begin{cases} 0 & ,if\ i = 0\ or\ j = 0 \\ 1 + LCS[i-1, j-1] & ,if\ \ p_i = Q_j \\ \max(LCS[i, j-1], LCS[i-1, j]) & ,p_i \neq Q_j \end{cases}$$

LCS[1,1] ➔ i= 1, j=1, $p_i = A$, Q_i=B

LCS[1,1]=1 LCS[2,1]=1 LCS[3,1]=1 LCS[4,1]=1 LCS[5,1]=2

LCS[1,2]=1 LCS[2,2]=1 LCS[3,2]=2 LCS[4,2]=2 LCS[5,2]=2

LCS[1,3]=1 LCS[2,3]=1 LCS[3,3]=2 LCS[4,3]=3 LCS[5,3]=2

LCS[1, 4]=1 LCS[2,4]=1 LCS[3,4]=2 LCS[4,4]=3 LCS[5,4]=2

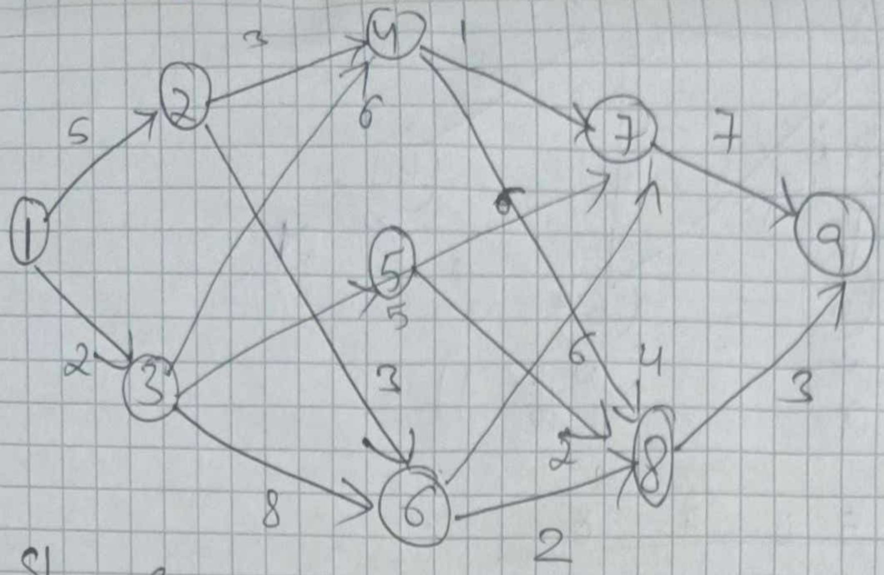
LCS[1,5]=2 LCS[2,5]=2 LCS[3,5]=2 LCS[4,5]=3 LCS[5,5]=2

LCS[1,6]=2 LCS[2,6]=3 LCS[3,6]=2 LCS[4,6]=3 LCS[5,6]=2

p_i

Q_i ⟶

			1	**2**	**3**	**4**	**5**	**6**
			a	**b**	**c**	**d**	**a**	**F**
		0	0	0	0	0	0	0
1	**a**	0	1	1	1	1	2	2
2	**c**	0	1	1	1	1	2	2
3	**b**	0	1	2	2	2	2	2
4	**c**	0	1	2	3	3	3	3
5	**f**	0	1	2	3	3	3	4



12
10
8
3

3 3

12+10+8+3.

S1 S2 S3 S4 S5

	1	2	2	3	3	3	4	4	5
	1	2	3	4	5	6	7	8	9
cost	12	8	10	7	5	5	7	3	0
dest	3	6	5	8	8	8	9	9	9

1 → 3 → 5 → 8 → 9

1 → 2 → 5 → 8 → 9

1, 1 → 2, 2

1, 1 → 2, 3

3, 6 → 4, 7

3, 6 → 4, 8

cost(3, 6) = min { C(6, 7) + cost(4, 7),
C(6, 8) + cost(4, 8) }

= 6 + 7, 2 + 3 = 5.

5 + 8 = 13

2 + 10 = 12

12

3, 5 → 4, 7

3, 5 → 4, 8

6 + 7

2 + 3 = 5.

3, 4 → 4, 7

3, 4 → 4, 8

1 + 7 = 8.

4 + 3 = ~~7~~

2, 2 → 3, 4

2, 2 → 3, 6

3 + 7

3 + 5 = 8.

2, 3 → 3, 4

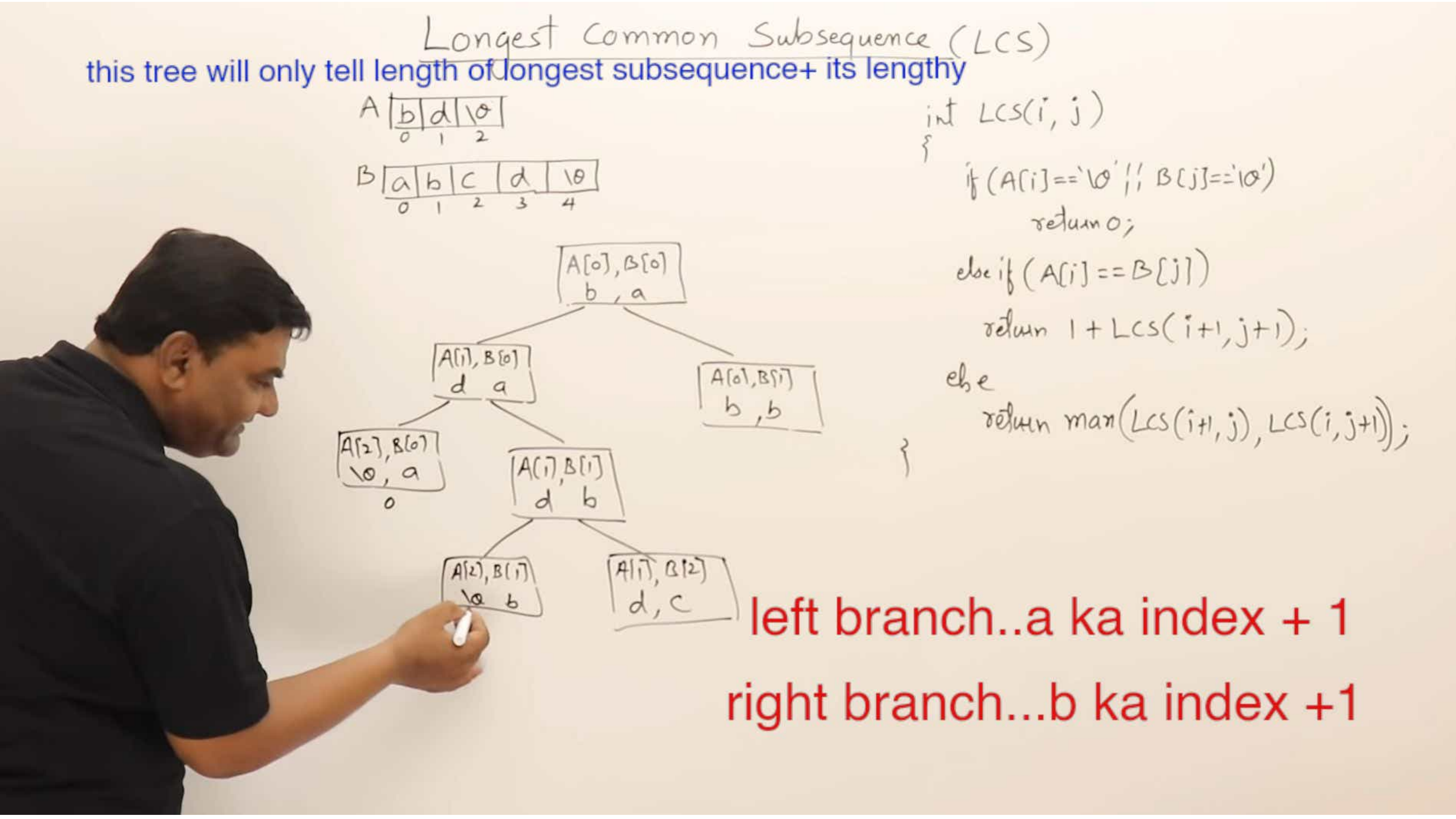
2, 3 → 3, 5

2, 3 → 3, 6

6 + 7 = 13

5 + 5 = 10

8 + 5 = 13 = 10


Longest Common Subsequence (LCS)
this tree will only tell length of longest subsequence+ its lengthy
int LCS(i, j)
{
if (A[i]=='\0' || B[j]=='\0')
return 0;
else if (A[i] == B[j])
return 1 + LCS(i+1, j+1);
else
return max(LCS(i+1, j), LCS(i, j+1));
}
left branch..a ka index + 1
right branch...b ka index +1