TOPIC NAME : Lab-1 DAY : Week-1

TIME : DATE : 03/12/24

Marks Distribution

"Do only what you can" ~ Sir

	Mid		Final	
	Week	%	Week	%
Offline	4, 7	4 %	8, 13	4 %
Online	2, 3, 5, 6	16 %	9, 10, 11, 12	16 %
Mid/Final	7	20 %	13	20 %
Perf.	1-7	10 %	8-14	~~50~~ 10 %
	Total	50 %	Total	50 %

→ Good Practice : Do it without any help at front

Offline : 8 % (4, 7, 8, 13) → ALWAYS Print if possible

Online : 32 % ~~(2 3,~~ (Almost Every week) ~~without~~

→ includes viva

Class Perf : 20 %

Exam (written) : 40 % (7, 14)

"GOOD LUCK"

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

Next Classes :

Online 1 (w-2) ~~2 (w-3)~~ : (From prev semester)

- Insertion Sort
- Quick Sort
- Merge Sort
- ~~Lean~~ Linear Search
- Binary Search.

} Do from home

↳ Explanation is necessary for ANY and ALL modifications

GOOD LUCK

TOPIC NAME : W-1 DAY : C-1

TIME : DATE : 3/12/24

Algorithm

Md. Khairul Hasan (Paris) Sir

01711109626

* Quiz 2 + Lab Mid ⟶ Same day
* Quiz 4 + Lab Final ⟶ "

"Just solve last semester question and CCI for A+"

~ Sir

"Lab is better for jobs" ~ Sir

* Books :- Thomas H. Cormen, Leiserson
  - Sartaj Sahni

GOOD LUCK™

TOPIC NAME : _____ DAY

TIME : DATE : / /

Algorithm:

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

"Infinity is something that cannot be reached"

Characteristics: * write lines

i) Input
ii) Output
iii) Definiteness
iv) Finiteness
v) Effectiveness

GOOD LUCK™

TOPIC NAME : W-1 DAY : C-2

TIME : DATE : 4/12/24

#Diff bet^n algo and program

[Check Slide]

#Slide Math

~~Given that,~~

(a)

For A,

10^{10} instructions is done in 1 s

1 " " $\frac{1}{10^{10}}$ s

$\therefore 2(10^7)^2$ " " $\frac{2\times10^{14}}{10^{10}}$ s

$= 2\times10^4$ s

$= 5.5$ hrs

For B,

10^7 instructions is done in 1 s

1 " " $\frac{1}{10^7}$ s

$\therefore 50\times10^7\log10^7$ " " $\frac{50\times10^7\log10^7}{10^7}$ s

$= 11.62.67$ s

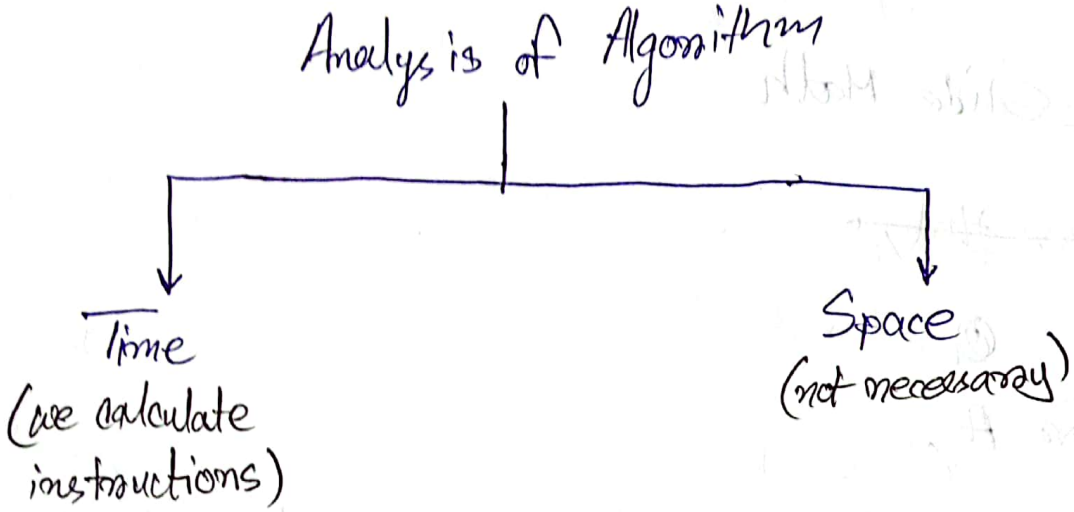
$\approx$ 20 mins

GOOD LUCK™

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

∴ B will finish first.



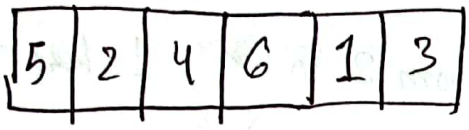
* C → constant that is used to compare
  └→ time constant

GOOD LUCK

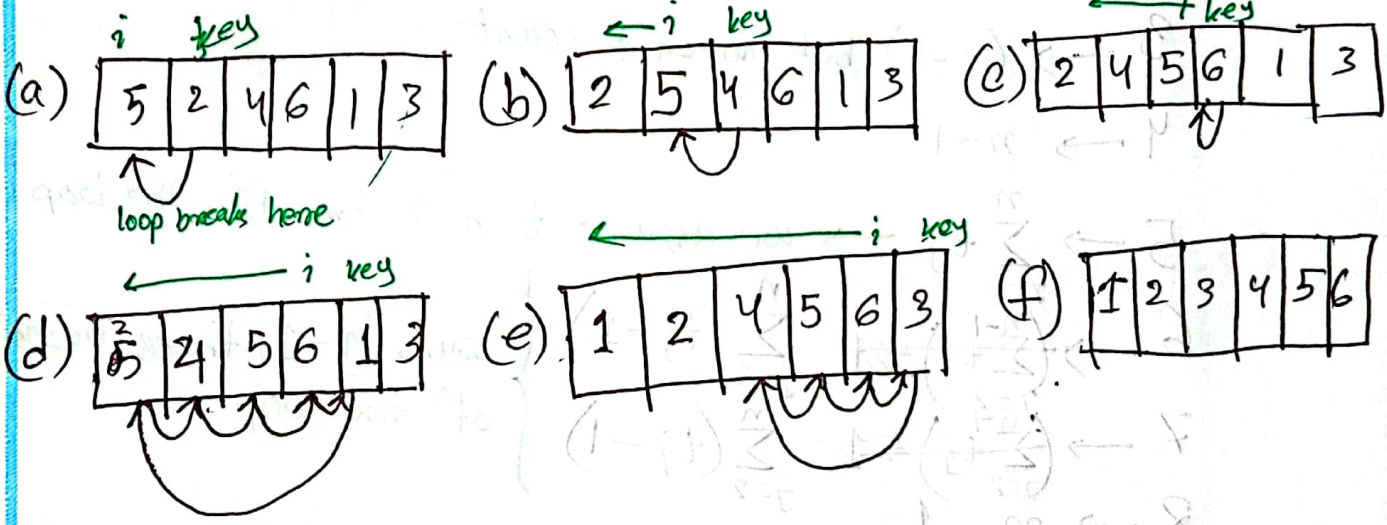
TOPIC NAME : W-1 DAY : C-3

TIME : DATE : 4 / 12 / 24

Insertion Sort



Magic happens in : while i > 0 and A[i] > key



* Frequency Count Method :

Using frequency to count how many times a line is executed

GOOD LUCK™

*There should be a proper value for each step. FIND IT

TOPIC NAME : _____ DAY _____

TIME : DATE : / /

[Insertion Sort] : Check Slide

j runs from 2 to n

1. ⟶ $n-1+1$ ⟶ to check if line is true

2 ⟶ $n-1$ ⟶ j runs from 2 to n ; 1 less than loop

3 ⟶ $(n-1)$ but doesn't count

4 ⟶ $n-1$

5 ⟶ $\sum_{j=2}^{n} t_j$ ⟶ we assume t as the time for the loop

6 ⟶ ~~$\left(\sum_{j=2}^{n-1} t_j\right)$~~ $\sum_{j=2}^{n} (t_j - 1)$

7 ⟶ ~~$\left(\sum_{j=2}^{n-1} t_j\right)$~~ $\sum_{j=2}^{n} (t_j - 1)$

} runs $(n-1)$ times version of the loop

8 ⟶ $n-1$

$\therefore$ Total Cost, $T(n) = c_1(n) + c_2(n-1) + 0 + c_4(n-1)$

$+ c_5\left(\sum_{j=2}^{n} t_j\right) + c_6\left\{\sum_{j=2}^{n} (t_j - 1)\right\} + c_7\left\{\sum_{j=2}^{n} (t_j - 1)\right\}$

$+ c_8(n-1)$

$= c_1(n) + (n-1)\{c_2 + c_4 + c_8\} + \sum_{j=2}^{n} (t_j) \times c_5$

$+ \sum_{j=2}^{n} (t_j - 1)(c_6 + c_7)$

GOOD LUCK™

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

Best Case Scenario : Sorted

∴ In best case, $t_j = 1$.

$$\therefore T(n) = nc_1 + (n-1)\{c_2 + c_4 + c_8\} + \sum_{j=2}^{n}(1)c_5$$

$$\sum_{j=2}^{n}(1-1)(c_6 + c_7)$$

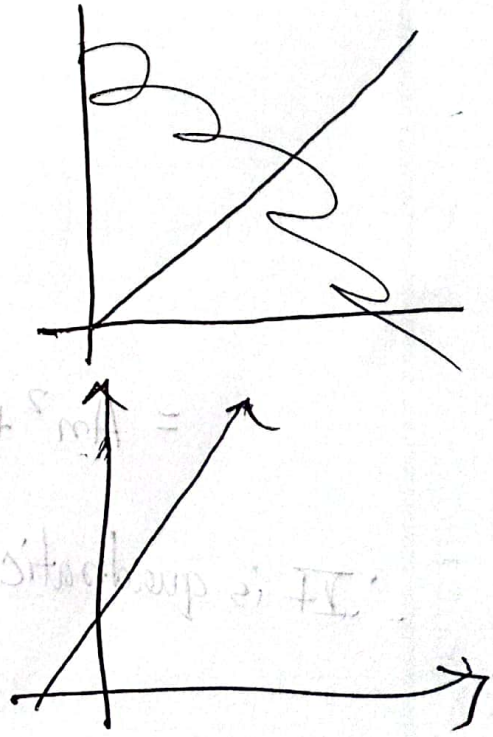
$$= nc_1 + (n-1)\{c_2 + c_4 + c_8\} + (n-1)c_5$$

$$= n(c_1 + c_2 + c_4 + c_5 + c_8) - (c_2 + c_4 + c_5 + c_8)$$

$$= An + B$$

$$= 0$$

∴ T(n) for best case will be a linear equation



GOOD LUCK™

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

Worst Case Scenario : Reverse sorted

$\therefore t_j = j$

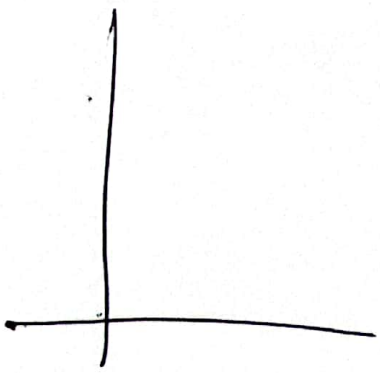
$$\therefore T(n) = nc_1 + (n-1)(c_2+c_4+c_8) + \sum_{j=2}^{n}(j)\,c_5 + \sum_{j=2}^{n}(j-1)(c_6+c_7)$$

$$= nc_1 + (n-1)(c_2+c_4+c_8) + \left(\frac{n(n+1)}{2} - 1\right)c_5 + \left(\frac{n(n-1)}{2}\right)(c_6+c_7)$$

$=$

$= An^2 + Bn + C$

$\therefore$ It is quadratic eq^n



GOOD LUCK

TOPIC NAME : ~~L-2 W-2~~ DAY : L-2

TIME : DATE : 10/12/24

Stable sort :

~~They~~

The type of sorting where previous sorting remains through current sorting

* what SHOULDNT be applied >> what SHOULD be applied

Online 2 :

BFS, DFS

TOPIC NAME: M-2 DAY: C-3 TIME: DATE: 16/12/24

Insertion Sort

Avg case = Half sorted, half unsorted
↳ In AT read before quiz

$T(n) =$ ⋮

$= an^2 + bn + c$

* Always say worst case when asked
* Take leading term when dealing with leading exponent

GOOD LUCK

//Make a merge sort visualizer
→ tree
→ push eqn in as string
→ inorder traversal

TOPIC NAME : _____ DAY : DATE : / /

Merge Sort

For Divide and Conquer:

$$T(n) = \begin{cases} \theta(1) & \text{if } n \le c \\ aT\left(\frac{n}{b}\right) + D(n) + C(n) & \text{else} \end{cases}$$

$$T(n) = \underbrace{D(n)}_{\text{for divide}} + \underbrace{aT(n/b)}_{\text{for conquer}} + \underbrace{C(n)}_{\text{for combine}}$$

for conquer
↓
number of sub problems = a
time " solving sub problems = $T\left(\frac{n}{b}\right)$

GOOD LUCK™

TOPIC NAME: ~~L-3~~ W-3 DAY: L-3

TIME: DATE: 17 / 12 / 24

Offline:

Title: Topological Sort and DAG Shortest Paths

* If !DAG → print Topological Sort

* Input: a DAG with weight (must have weight +-ve)

* Output: any node will not be able to go left

Shortest Path:

* Inp: Node

* Output: All paths with their cost

↳ only the lowest is counted.

GOOD LUCK™

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

For each loop :⟶

for (int/auto num : structure)

{

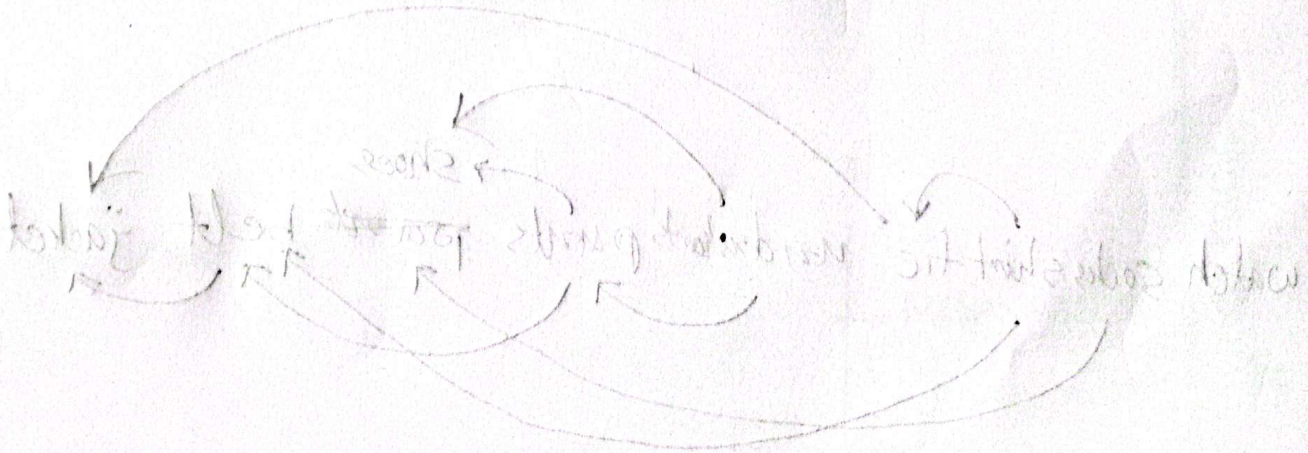
do what you want

}

*iterator : ig it points to a number

v.begin() ⟶ starting position

v.front() ⟶ " " value



GOOD LUCK™

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

For each loop :⟶

for (int/auto num : structure)

{

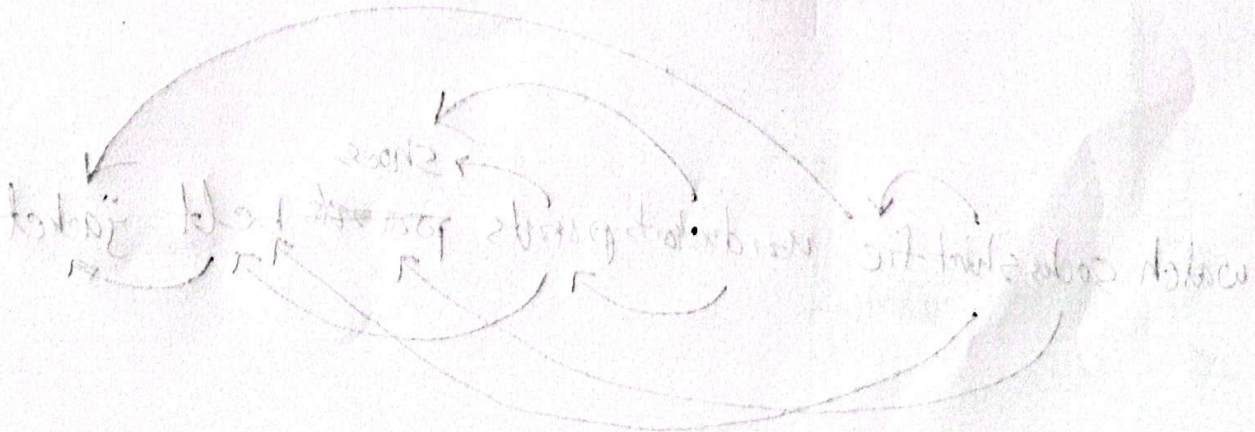
do what you want

}

*iterator : ig it points to a number

v.begin() ⟶ starting position

v.front() ⟶ " " value



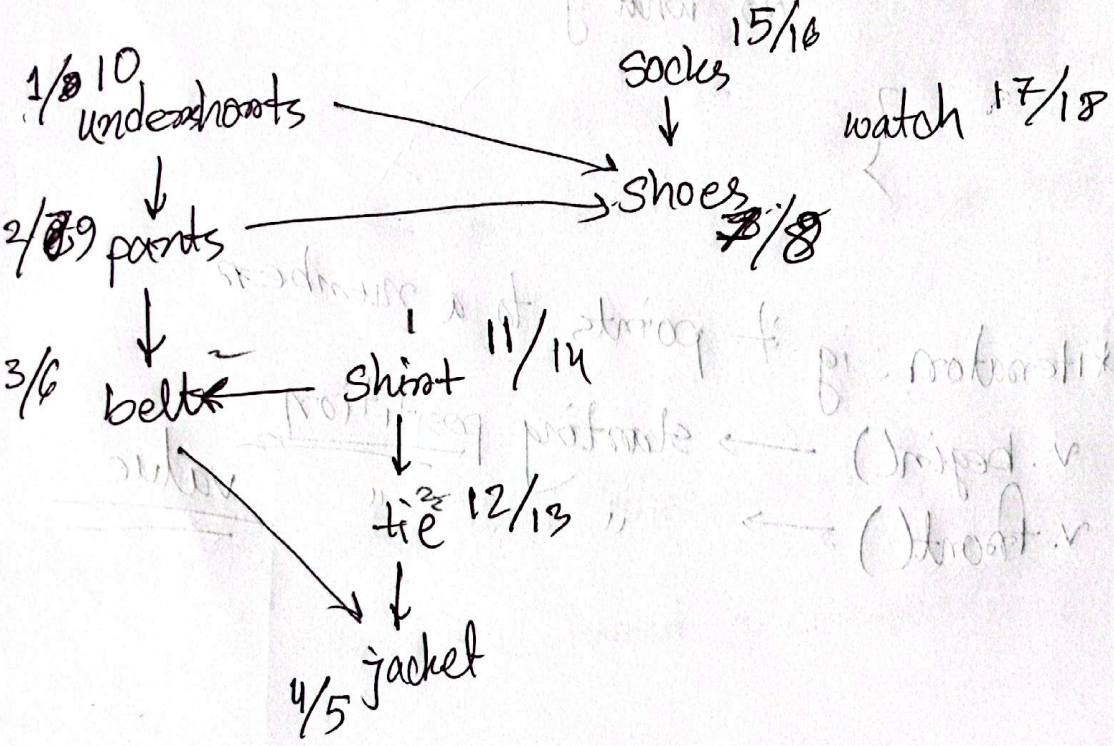
GOOD LUCK™

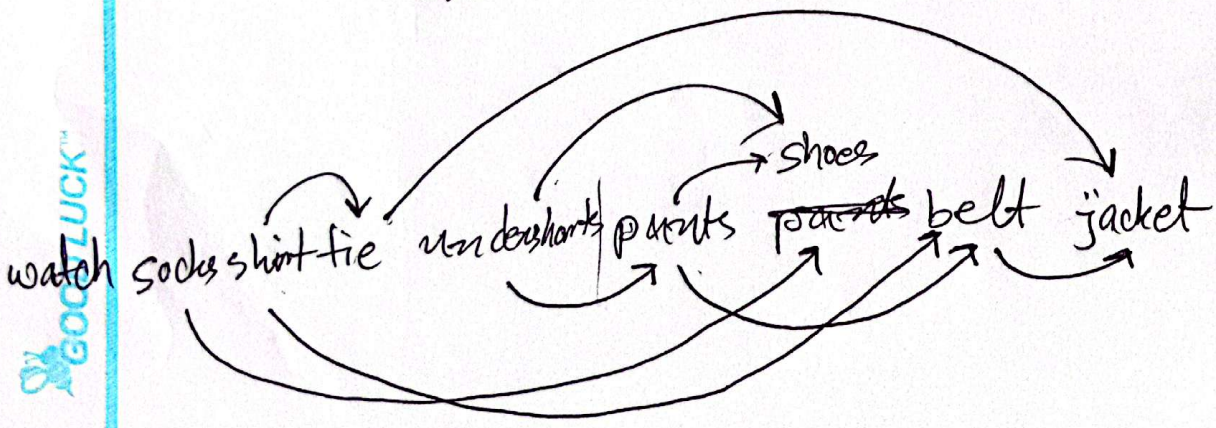
TOPIC NAME : W-3 DAY : C-6

TIME : DATE : 17/12/24

Topological Sort

//Solve Einstein's riddle thingy with Topological sort





GOOD LUCK

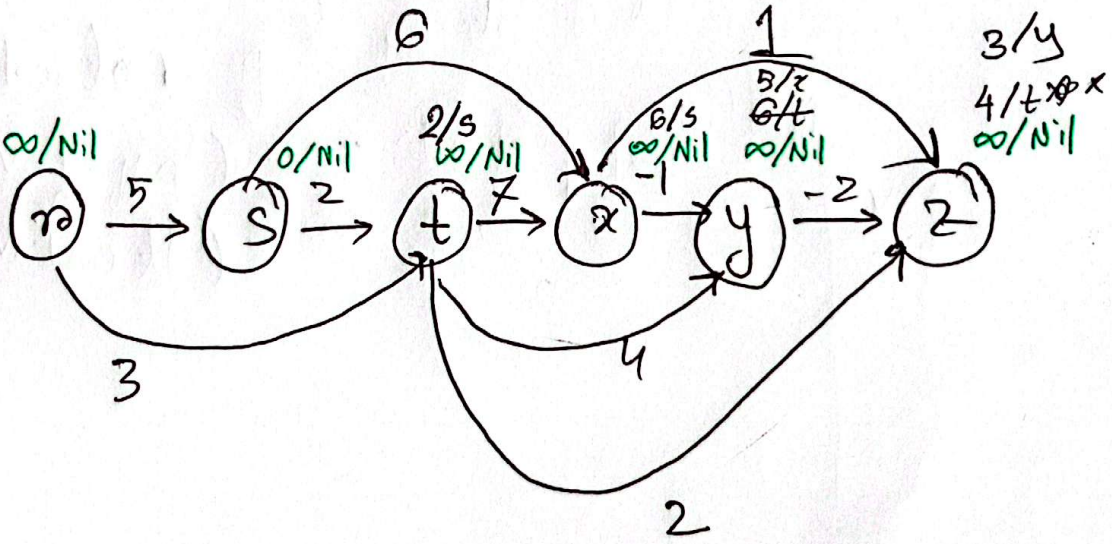
TOPIC NAME : W-3 DAY : C-7

TIME : DATE : 8/12/24

Single Source Shortest Path :

Inp : Directed weighted graph

Outp : ~~Sho~~ Shortest Cheapest path to every other node



S :

~~r : ∞~~

r : ∞

s → t : 2

s → x : 6

s → x → y : 5

s → x → y → z : 3

→ Make table from slide

GOOD LUCK™

TOPIC NAME: U-4 DAY: L-4

TIME: DATE: 24/12/24

Travelling Salesman Problem using Branch and Bound

State Space Tree

Solutions represented in a tree

- each node can be 3 states
  - Live node: Until children are analyzed
  - Expanding node: When children are being analyzed
  - Dead node: When no more children can be analyzed

Branch and Bound

Used in optimization problems

↓

break problems into branches to solve them

- (1) FIFO Branch and Bound
- (2) LIFO "
- (3) Least cost "

GOOD LUCK

TOPIC NAME : DAY : TIME : DATE : / /

TSP: Visit every city (node) exactly once to reach back to the first place (node) using least cost

Branch and Bound

* Set cost of each node to itself as ∞
* Each node will cost min will be AT LEAST (enter + leave) (min)
* * Set Add cost to be (leave − min(leave))

a → b = 8
a → c = 9
} a → c = 1

* Find min in each row of adjacency matrix
* * reduce row wise as leave − min
} For leaving

* Find min in each column of the REDUCED matrix
* * Reduce column wise as entry − min
For entering

* Sum of (Sum of row min) and (Sum of column min)
= Min traveling cost → 38

GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

* When taking i to j, make the i row and j-column inf and (j, i) inf.

** if matrix can't be reduced further $\hat{r} = 0$

** else, $\hat{r}$ = New Min row + New Min Column

** If $i \to j$; $\Rightarrow j$ = cost(i, j) + r + $\hat{r}$

GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

INF	10	17	0	1
12	INF	11	2	0
0	3	INF	0	2
15	3	12	INF	0
11	0	0	12	INF

$1 \rightarrow 5$

INF	INF	INF	INF	INF
12	INF	11	2	INF → 2
0	3	INF	0	INF → 0
15	3	12	INF	INF → 3
INF	0	0	12	INF → 0
↓	↓	↓	↓	
0	0	0	0	

$\therefore \hat{n} = 5$

$\therefore \text{cost}(1 \rightarrow 5) = 1 + 25 + 5$

$= 31$

GOOD LUCK

TOPIC NAME : DAY :

TIME : DATE : / /

1 → 4 → 3

	1	2	3	4	5
1	INF	INF	INF	INF	INF
2	12	INF	INF	INF	0 → 0
3	INF	3	INF	INF	2 → 2
4	INF	INF	INF	INF	0 → 0
5	11	0	INF	INF	INF → 0
	↓	↓	↓	↓	↓
	11	0	0		0

$\hat{n} = 2 + 11 = 13$

$= (12 + 25) + 25 + 13$

$= 63\ 50$

"GOOD LUCK"

TOPIC NAME

	1	2	3	4	5	
1	∞	∞	∞	∞	∞	1 → 4 → 2 → 5
2	∞	∞	28	∞	∞	
3	0	∞	∞	∞	∞ → 0	
4	∞	∞	8	∞	∞ → 0	
5	∞	∞	∞	∞	∞	

1 → 0

cost = 0 + 28 + 0 = 28

#Offline on the 13th Week

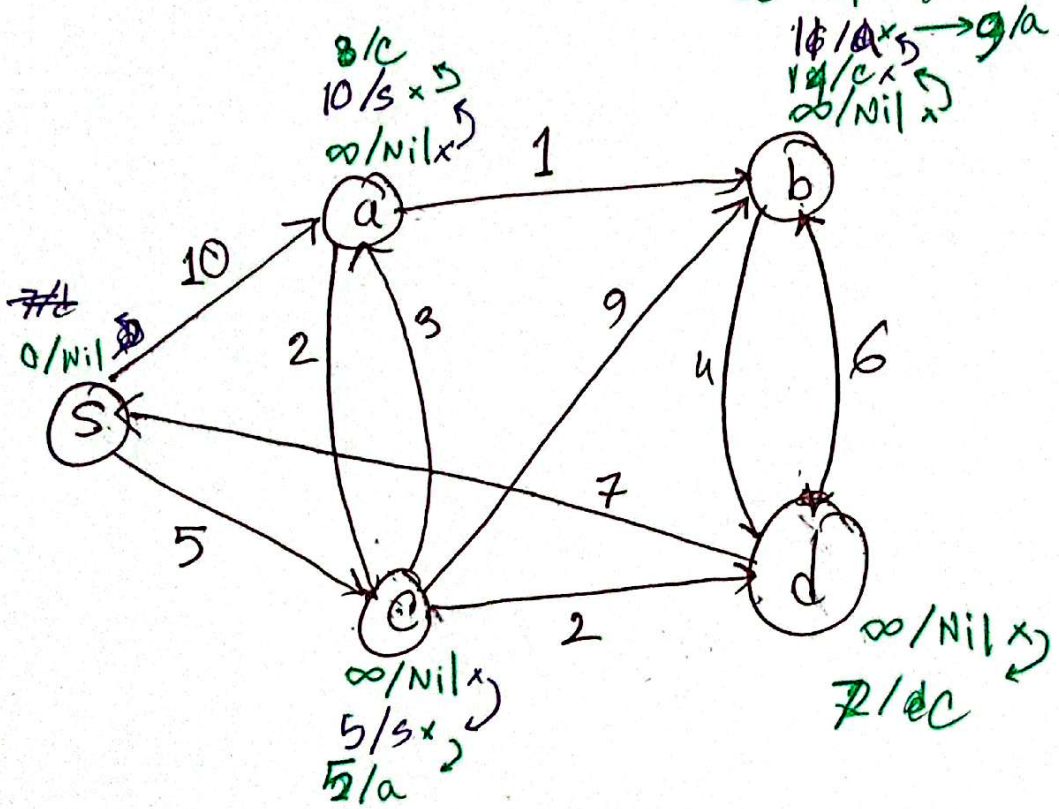
#Online - 3 : i) Dijkstra
Single Source shortest path ii) Bellman Ford

TOPIC NAME : Algo W-3 DAY : TIME : DATE : 19/12/24

Q. Page - 645

Dijkstra

∴ The graph has cycle, it cannot be topologically sorted



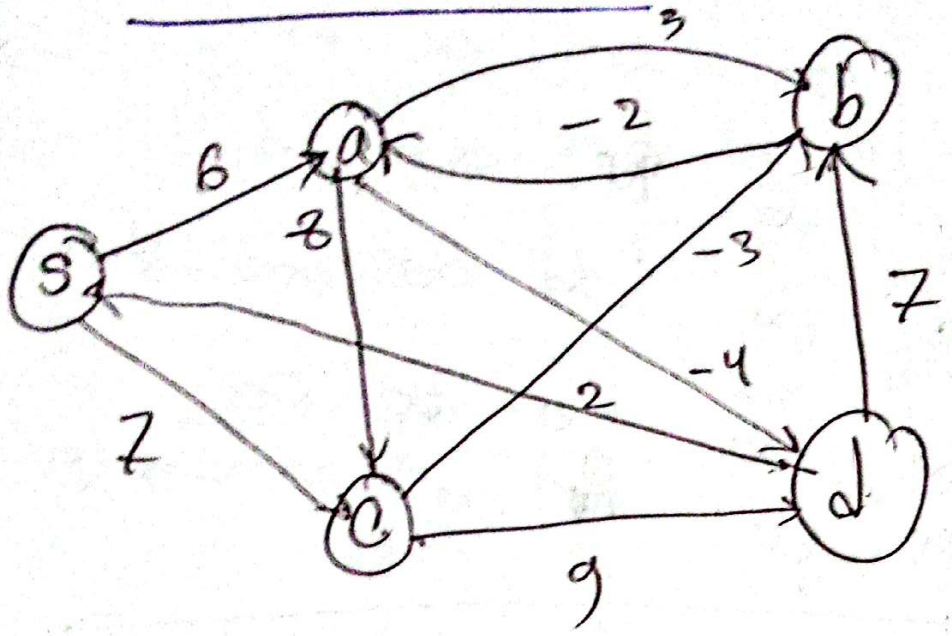
selected Node		s	a	b	c	d
Initialize		~~∞~~ 0/Nil	∞/Nil	∞/Nil	∞/Nil	∞/Nil
1	s	0/Nil	~~10/s~~ 10/s	∞/Nil	5/s	∞/Nil
4	a	0/Nil	~~10/s~~ 8/c	~~9~~/a	5/s	~~∞/Nil~~ 7/c
5	b	0/Nil	8/c	9/a	5/s	~~7~~ 7/c
2	c	0/Nil	8/c	~~13~~/c	5/s	7/c
3	d	0/Nil	8/c	13/d	~~8~~ 5/s	7/c

GOOD LUCK

TOPIC NAME: W-4 DAY: C-8

TIME: DATE: 24/12/24

Bellman-Ford



Edge Order	Cost/Weight
(a, b)	5
(a, c)	8
(a, d)	-4
(b, a)	-2
(c, b)	-3
(c, d)	9
(d, s)	2
(d, b)	7
(s, a)	6
(s, c)	7

GOOD LUCK

TOPIC NAME : DAY
TIME : DATE : / /

Pass No. →	s	a	b	c	d
Initialize	0/Nil	∞/Nil	∞/Nil	∞/Nil	∞/Nil
1	0/Nil	~~6/s ∞/Nil~~	~~∞/Nil ∞/Nil~~	~~7/s ∞/Nil~~	~~∞/Nil ∞/Nil~~
2	0/Nil	~~6/s 9/b~~	~~9/d 11/a~~	~~7/s 11/a~~	2/a
3	0/Nil	2/b	4/c	7/s	2/a
4	0/Nil	2/b	4/c	7/s	-2/a
5	0/Nil	2/b	4/c	7/s	-2/a

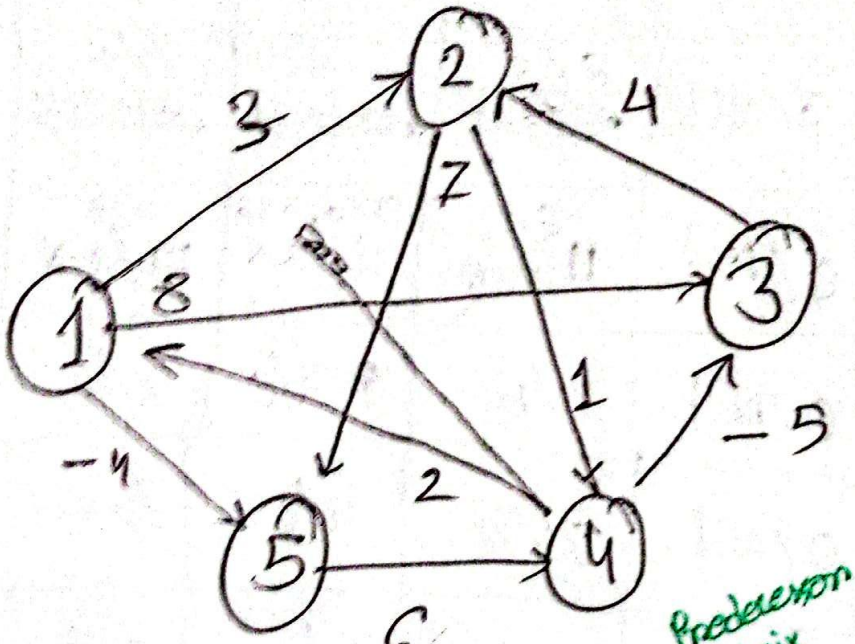
do it (node-1) times

GOOD LUCK

TOPIC NAME: W-4 DAY: C-9

TIME: DATE: 26/12/24

All pair Shortest path.



Pointers Matrix

$$D_0^0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \end{array}$$

Predecessor Matrix

$$\pi_0^0 = \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} Nil & 1 & 1 & Nil & 1 \\ Nil & Nil & Nil & 2 & 2 \\ Nil & 3 & Nil & Nil & Nil \\ 4 & Nil & 4 & Nil & Nil \\ Nil & Nil & Nil & 5 & Nil \end{bmatrix} \end{array}$$

$$d_{42}^{(1)} = \min \begin{cases} d_{42}^{(0)} \\ d_{41}^{(0)} + d_{12}^{(0)} \end{cases}$$

* diagonal and for D^k, nth row and nth column will not change

TOPIC NAME : DAY :

TIME : DATE : / /

$$D^1 = \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & \infty \\ \infty & 4 & 0 & \infty & \infty \\ 2 & -5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \qquad \Pi^1 = \begin{bmatrix} Nil & 1 & 1 & Nil & 1 \\ Nil & Nil & Nil & 2 & 2 \\ Nil & 3 & Nil & Nil & Nil \\ 4 & 1 & 4 & Nil & 1 \\ Nil & Nil & Nil & Nil & 5 \end{bmatrix}$$

d_{54}

* $\Pi_{ij}^{(n)}$ will be replaced by Π_{nj}

$d_{11}^{(1)} = d_{11}^{(0)} + d_{11}^{(0)}$

$= 0 + 0 = 0$

$d_{12}^{(1)} = d_{11}^{(0)} + d_{12}^{(0)}$

$= 0 + 3 = 3$

$d_{54}^{(1)} = d_{51}^{(0)} + d_{14}^{(0)}$

$= \infty + \infty = \infty$

$d_{14}^{(2)} = d_{12}^{(1)} + d_{24}^{(1)}$

$= 3 + 1 = 4$

$d_{34}^{(2)} = d_{32}^{(2)} + d_{24}^{(2)} = 4 + 1 = 5$

$d_{(12)}^{(5)} = d_{15}^{(4)} + d_{52}^{(4)}$

$= -4 + 5 = 1$

$D^2 =$ [

GOOD LUCK

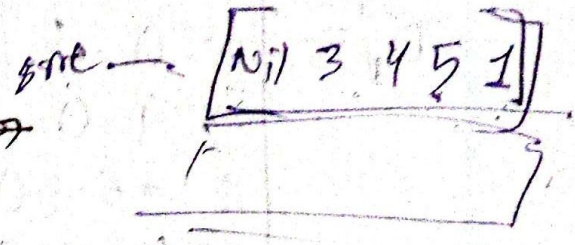
TOPIC NAME : DAY :

TIME : DATE : / /

1 → 4

Path : 1 → 4 ← 5 ← 3 ← 1

Cost : 2



1 → 5 → 4

1 → 3

Path : 3 ← 4 ← 5 ← 1

Cost :

Slide Algo :

* W = Weight Matrix

* For predecessor, we if condition

$$\min\left(d_{ij}^{(k-1)}, \left(d_{ik-1}^{(k-1)} + d_{k-1\,j}^{(k-1)}\right)\right)$$

if $(d_{ij}^{k} != d_{ij}^{k-1})$

$$\pi_{ij}^{k} = \pi_{nj}^{k-1}$$

* Guess of why it updates :

Dⁿ updates bc pre n checks if shortest path to d_{ij}^{n} comes from n

GOOD LUCK

TOPIC NAME : W-45 DAY : L-5

TIME : DATE : 31/12/24

Online - 4

Floyd Warshall

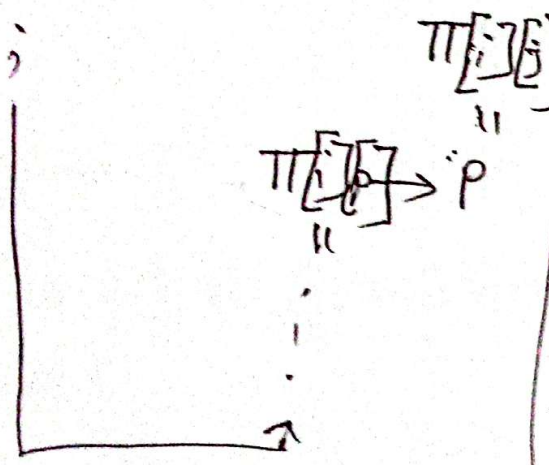
GOOD LUCK

TOPIC NAME: W_5 DAY: C-10

TIME: DATE: 02/01/25

Floyd Warshall

For $i \rightarrow \ldots \rightarrow j$,



In order to ~~check~~ visit the ~~j~~ dest ~~index~~, we need to visit pre[src][dest]

↳ Use github print path

*FLOYD WARSHALL simulates each nodes to check if all prev nodes can shorten distance

d_{ij}^{k} → implies all ~~pre~~ nodes in 1 to k

GOOD LUCK

TOPIC NAME : _____ DAY : _____

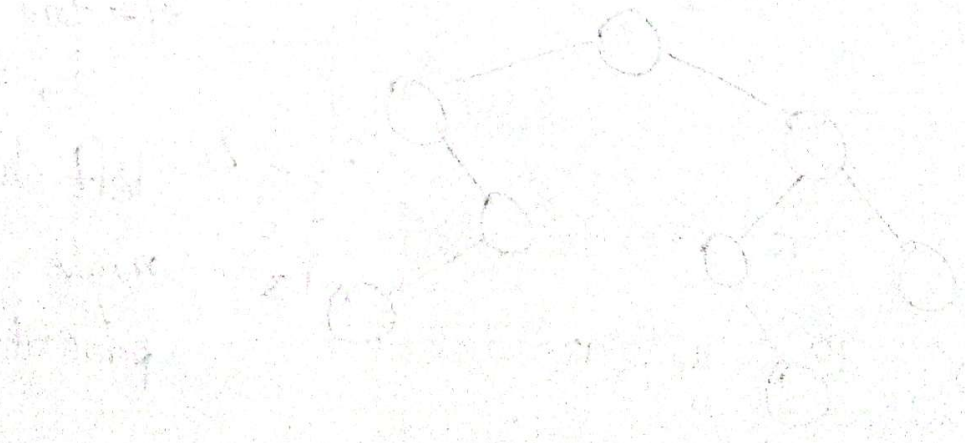
TIME : DATE : / /

For k=1,

i ⟶ . . . . . ⟶ j

For k, if cost decreases:

i ⟶ . . . ⟶ k ⟶ . . . ⟶ j



GOOD LUCK™

TOPIC NAME : W-6 DAY : L-6

TIME : DATE : 7 / 1 / 25

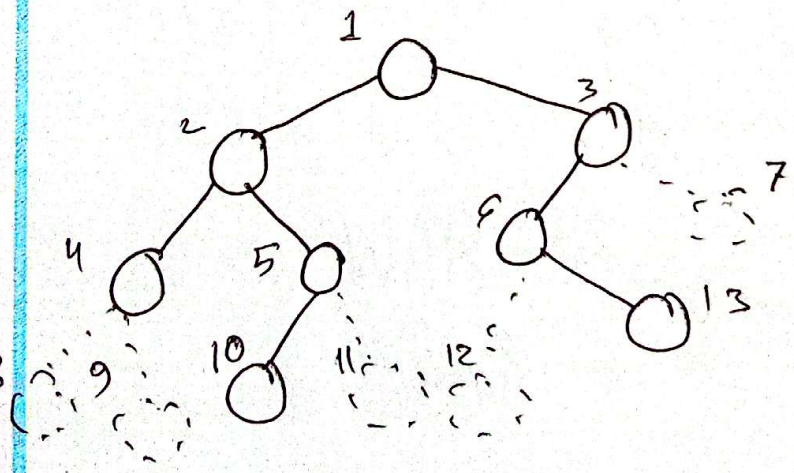
Offline – 2 ;

Johnson's Algorithm

Heap: A data structure that

- is ~~written as a~~ expressed as a complete binary tree
  - Must have two children and must be filled from left
  - 1. Must complete a level
  - 2. If level completed, inserted from left
- is coded with array

For a BINARY Tree :



~~If~~ for any node i, the index;

left child = i * 2

right " = (i * 2) + 1

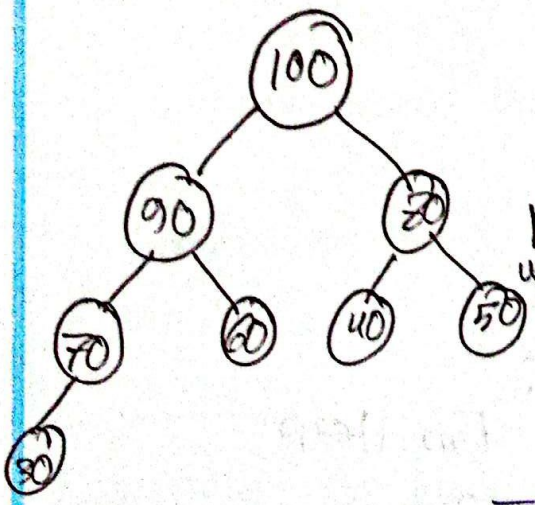
parent = floor(i/2)

TOPIC NAME : DAY :
TIME : DATE : / /

For Heap:

Max Heap → Nodes >= Child
Min " → Nodes <= Child

A max-heap:



Insertion:

i) Add in the usual spot
ii) Compare with parent
loop until → if (parent < data)
swap
→ else
win

Max complexity: $\log_2 n$

: For inserting/deleting n number of data:
Complexity = $n \log n$

Delete:

*) We can ~~only~~ ONLY deleteroot
i) Shift lowest node to root
ii) Compare with children
→ ~~if~~ swap with highest child
until → child < data

GOOD LUCK

TOPIC NAME : _____ DAY : _____ TIME : DATE : / /

Heapify:

* Assume leaf is ~~son~~ in its position

i) From half of array: ~~compare with children~~

compare with children

if ~~smallest~~ child > data,

swap

until data > child

Priority Queue

If we take

	For Array	For Heap
push()	1	$\log_2 n$
then pop()	n	$\log_2 n$

GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

Heap sort:

i) Make the DS a max-heap

ii) ~~A~~ The root will ALWAYS be max

iii) For n data :

~~inse~~ remove root and save it

heapify()

insert the prev root at nth index

n--

until n = 0

Lab Mid : 20 Marks :

5 MCQs

1 Algo

1 Example/Scenario

GOOD LUCK

TOPIC NAME: 10-6 DAY: C-11 DATE: 7/1/25

Spanning Tree and Minimum Spanning Tree

Tree:

Connected Acyclic Graph

* n nodes have n-1 edges

* Tree with the lowest weight is minimum spanning tree.

Kruskal's Algorithm

$T = \{(h,g)_1, (i,c)_2, (g,f)_2, (a,b)_4, (b,f)_4, \cancel{(i,g)}, (c,d)_7, \cancel{(i,h)}, (a,h)_8, \cancel{(b,c)}, (d,e)_9, \cancel{(b,h)}, \cancel{(f,e)}, \cancel{(d,f)}\}$

Cost : $1 + 2 + 2 + 4 + 4 + 7 + 8 + 9$

$= 37$

GOOD LUCK

TOPIC NAME : DAY :

TIME : DATE : / /

[1] Safe edge :

Safe edge wiki

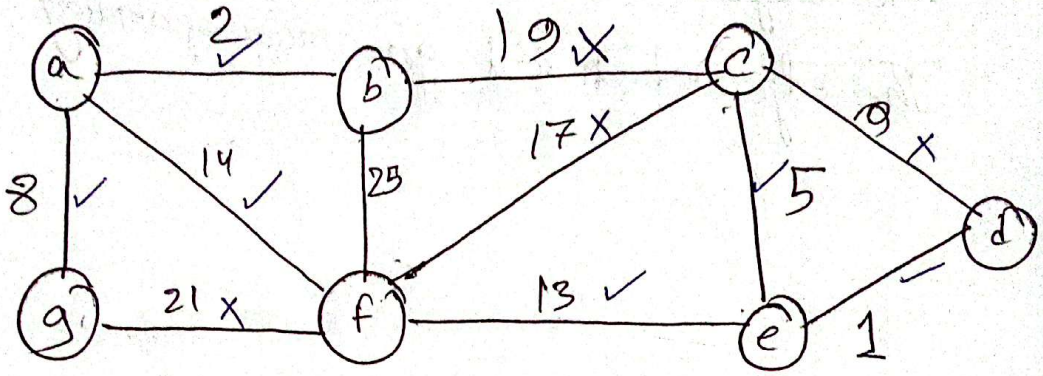
The edge which will give min spanning tree

GOOD LUCK

TOPIC NAME : W-6 DAY : C-12

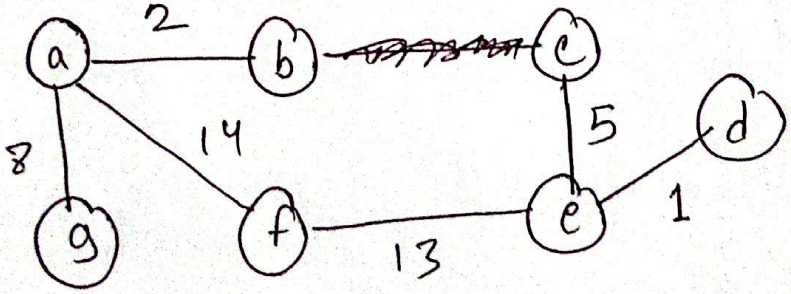
TIME : DATE : 9/1/25

Kruskal's Algorithm



$T = \{(d,e), (a,b), (c,e), (a,g), \cancel{(c,d)}, (f,e), (a,f), \cancel{(f,c)}, \cancel{(b,c)}, \cancel{(g,f)}\}$

Cost : 1 + 2 + 5 + 13 + 14 = 43



GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

Safe edge : .

Cut : Partition of vertices

Cross : If edge connects two vertices on two ~~point~~ partition, the edge crosses the cut

Respect : If the set ~~does~~ contains edges that DO NOT cross the cut, it IS NOT respecting.

Light Edge : Lowest edge weight in cross

MakeSet(v) :

Pg - 626 : Given Undirected weighted Graph and incomplete A (safe edge), cut which respects graph and (u, v) is light edge. Proove (u, v) is a safe edge

↳ Cut changes ! use proof by contradiction

GOOD LUCK

TOPIC NAME: W-7 DAY: L-7 TIME: DATE: 14/1/25

STL

pair <type1, type2> name = make_pairs(thing1, thing2)

↓

also works with just ( )

↓

access using name.first and name.second

map <type1, type2> name;

name[key] = value → access ONLY value using key

↳ also works with m.insert({key, value})

↳ keys are unique but replaceable

GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

Online - 5:

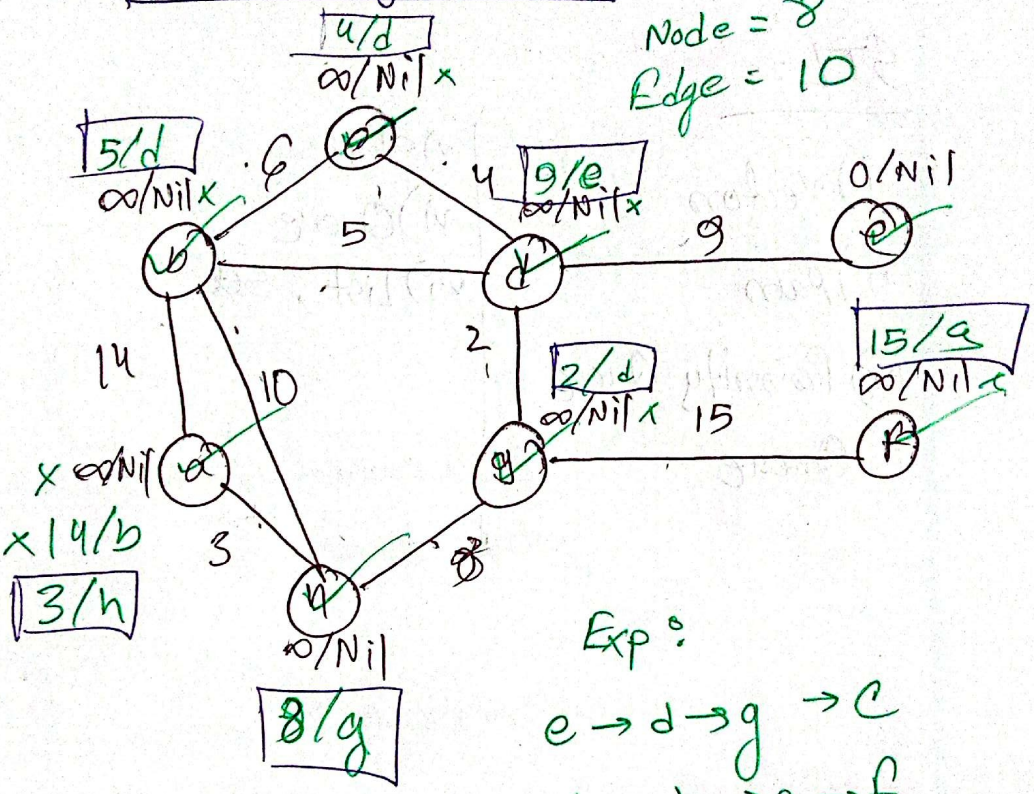
STL:

i) Vectors
ii) Pair
iii) Priority Queue
iv) Queue
v) Stack
vi) Queue
vii) List, Set

TOPIC NAME : W-7 DAY : C-13

TIME : DATE : 14 / 1 / 25

Prim's Algorithm



Exp :

e → d → g → c

→ b → h → a → f

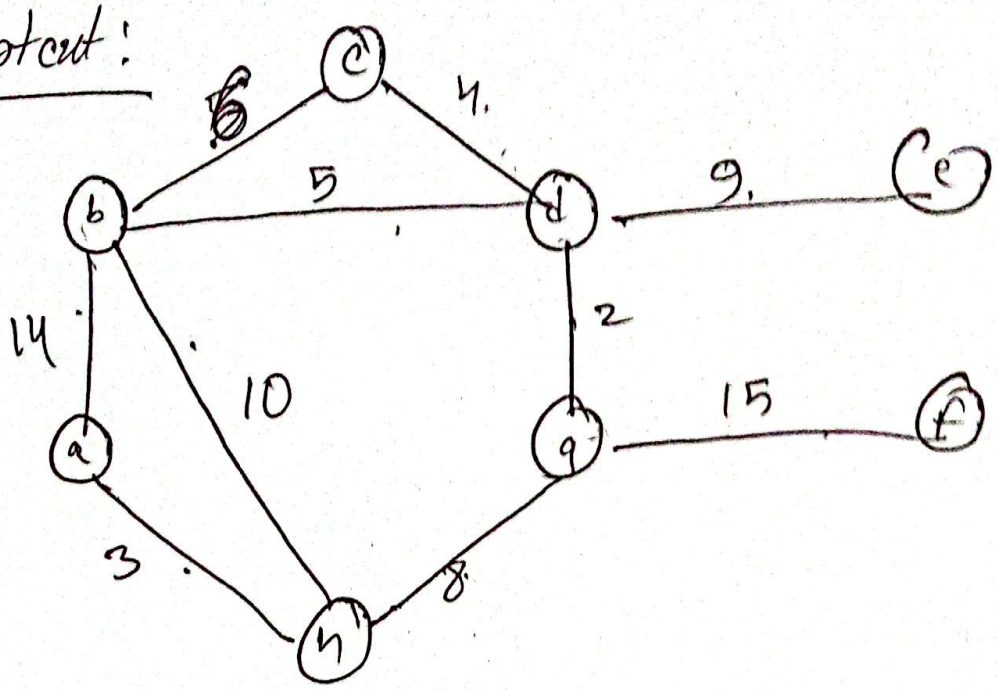
$$T=\left\{\frac{(d,e)}{9},\frac{(c,d)}{4},\frac{(f,g)}{15},\frac{(g,d)}{2},\frac{(b,d)}{5},\frac{(h,g)}{8},\frac{(a,h)}{3}\right\}$$

GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

Shortcut :



i) Cut the graph keeping one vertex, V_0 one side

ii) Take lightest edge with the $V_0 \rightarrow V_0 - V_1$

iii) Cut with V_0, V_1 Push the edge

iv) Loop

GOOD LUCK™

TOPIC NAME: 10-7 DAY: C-14

TIME: DATE: 15/1/23

Solving Recurrences – Pg 83

Substitution: $T(n) = 2T(\lfloor n/2 \rfloor) + n$

~~Let,~~

~~the solution~~

Guess that, $T(n) = O(n \lg n)$

$T(n) \le cn \lg n$ → as per the conditions of Big O

We assume that, ~~$T(n/2)$~~ $T(\lfloor \frac{n}{2} \rfloor) \le c\lfloor \frac{n}{2} \rfloor \lg \lfloor \frac{n}{2} \rfloor$ is true

↳ We dont know if its equal or lesser

~~$T(n) \le 2T\left(\frac{n}{2}\right)$~~

$$T(n) \le 2\left(c\lfloor \frac{n}{2} \rfloor \lg \lfloor \frac{n}{2} \rfloor\right) + n$$

$$\le cn \lg\left(\frac{n}{2}\right) + n = cn \lg n - cn \lg 2 + n$$

$$= cn \lg n - cn + n$$

We skip this part which is why it makes it bigger

$\therefore T(n) \le cn \lg n$

GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

Master Method: → 95

* Master method can only solve recurrences of

$T(\frac{n}{b}) + f(n)$ $\boxed{aT(\frac{n}{b}) + f(n)}$

Three cases : → 93

i) if $f(n) = O(n^{\log_b a - \epsilon})$; for some constant $\epsilon > 0$

then $T(n) = \theta(n^{\log_b a})$

ii) if $f(n) = \theta(n^{\log_b a})$

then $T(n) = \theta(n^{\log_b a} \log n)$

iii) if $f(n) = \Omega(n^{\log_b a + \epsilon})$ and $af(\frac{n}{b}) \leq cf(n)$ for some constant $c < 1$

then $T(n) = \theta(f(n))$

GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

$T(n) = 9T\left(\frac{n}{3}\right) + n$ using Masters Method

Here,

$$n^{\log_b a} = n^{\log_3 9}$$

$$= n^{2\log_3 3}$$

$$= n^2$$

$$= n^{2+0}$$

which is first condition of Masters method

$$\therefore T(n) = \theta(n^{\log_b a}) = \theta(n^2)$$

GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

$T(n) = 4T\left(\frac{n}{2}\right) + n^3$

4

$4T\left(\frac{n}{2}\right)$

$4\left(\frac{n}{2}\right)^3$

$= 4 \times \frac{n^3}{8}$

$= \frac{n^3}{2} \leq cf(n)$

$\Theta = \frac{1}{2} n^3$

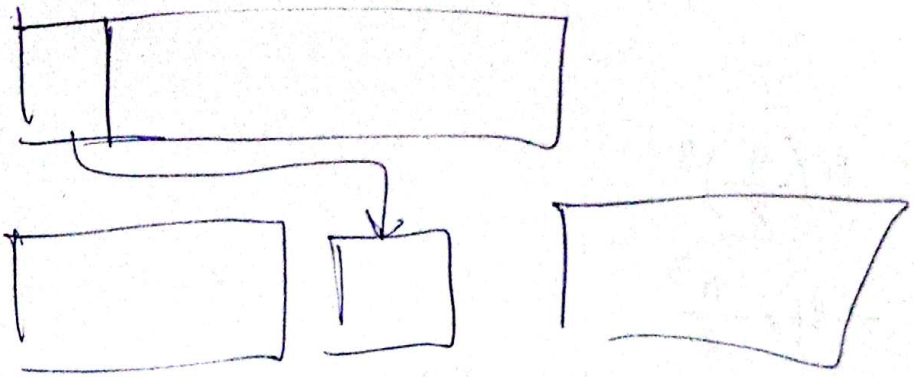
where $c = \frac{1}{2}$

GOOD LUCK

TOPIC NAME: W-7 DAY: C-15

TIME: DATE: 16/1/25

Quick Sort → 176



Partition ( )

p = Partition ( ) —— $\theta(n)$

Quicksort() —— $T(n)$ → $\frac{n}{2}$ for best, $n-1$ for worst

.. —— $T(\ )$ → $\frac{n}{2}$ for best, 1 for worst

Best Case: $T(n) = 2T(\frac{n}{2}) + \theta(n)$

Worst Case: $T(n) = T(n-1) + \theta(n)$ → Master method will not work

Avg Case: $T(n) = T(\frac{n}{10}) + T(\frac{9n}{90}) + \theta(n)$

$\frac{1}{10} \times n$

GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

Best case → 0 — 100

40 — 60

35 — 65

.

.

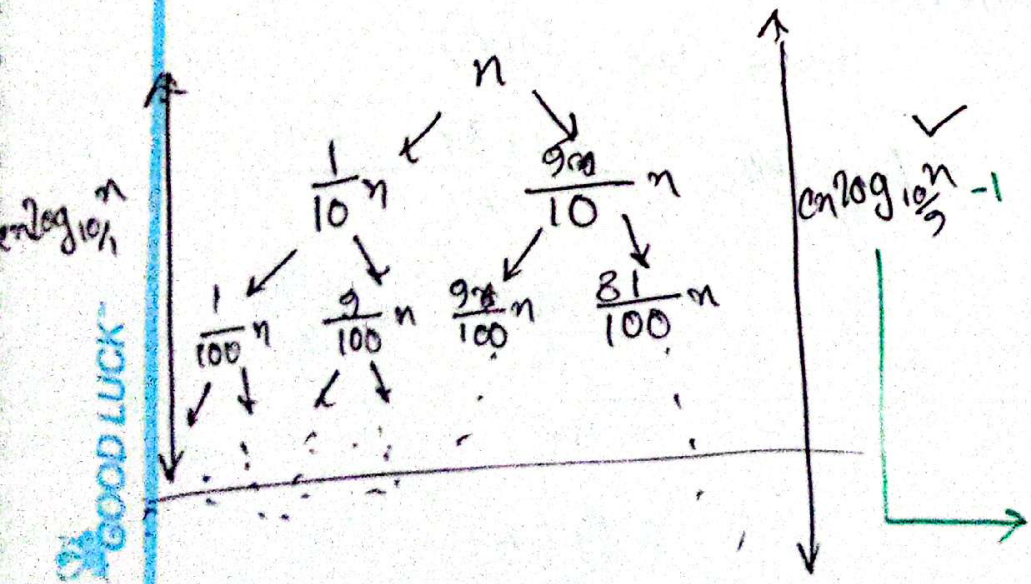
.

15 — 85

10 — 90

Worst case → 0 — 100

$\hookrightarrow T(n) = T(\frac{n}{10}) + T(\frac{9n}{10}) + cn$



$T(n/10) = T(\frac{n}{100}) + T(\frac{9n}{100})$

$T(9n/10) = T(\frac{9n}{100}) + T(\frac{81n}{100})$

for partitioning in a tree, ~~where~~ formula is $\log_{\text{total/child}} n$

GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

Now,

$$cn \log_{10/9} n = \left(cn \times \frac{\log_2 n}{\log_2 (10/9)} \right) \rightarrow \text{constant}$$

$$= \theta(n \log n) \rightarrow \text{Closer to Best case}$$

$$\therefore \text{Average Case} = \frac{\text{All cases}}{\text{Number of Cases}}$$

For linear search: Best Case : $\theta(1)$

Worst Case : $\theta(n)$

$$\text{Avg Case} : \frac{1+2+3+\cdots+n}{n}$$

$$= \frac{\frac{n(n+1)}{2}}{n} = \frac{n+1}{2}$$

$$= \theta(n)$$

GOOD LUCK

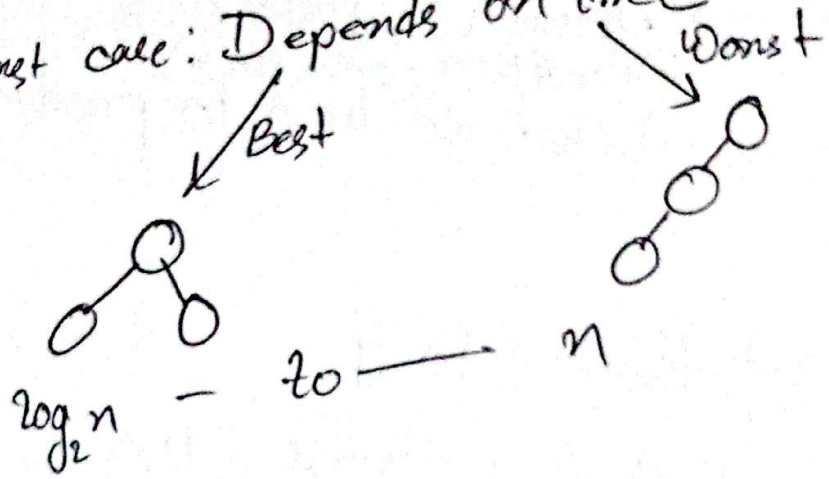
TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

For binary ~~tree~~ search :

Best case : $\theta(1)$

Worst case : Depends on tree



GOOD LUCK

TOPIC NAME : W-8 DAY : C-16

TIME : DATE : 23 / 1 / 25

TSP (using DP)

*Define → i) Given (a graph)
ii) Condition (only once)
iii) What we have to produce (shortest path)

C_{ij} = cost from i to j

$g(i, S)$ = length of a shortest path
starting from i
going through all vertices in S → set { }
and terminating at 1 → source

$g(7, \{5, 6\})$ → We are at 7
after visiting 5 and 6
we have to ~~vis~~ reach 1

$g(n, \{\}) = C_{n1}$

GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

Node	1	2	3	4
1	0	10	15	20
2	5	0	9	10
3	6	13	0	12
4	8	8	9	0

Visiting No Nodes

$\therefore g(2, \phi) = c_{21} = 5$

$\therefore g(3, \phi) = c_{31} = 6$

$\therefore g(4, \phi) = c_{41} = 8$

Visiting 1 Nodes:

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

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

$\therefore g(3, \{2\}) = c_{32} + g(2, \phi) = 13 + 5 = 18$

$\therefore g(3, \{4\}) = c_{34} + g(4, \phi) = 12 + 8 = 20$

$\therefore g(4, \{2\}) = c_{42} + g(2, \phi) = 8 + 5 = 13$

$\therefore g(4, \{3\}) = c_{43} + g(3, \phi) = 9 + 6 = 15$

Visiting 2 Nodes:

$\therefore g(2, \{3,4\}) = \min \begin{cases} c_{23} + g(3, \{4\}) = 9 + 20 = 29 \\ c_{24} + g(4, \{3\}) = 10 + 15 = 25 \end{cases} = 25$

$g(3, \{2,4\}) = \min \begin{cases} c_{32} + g(2, \{4\}) = 13 + 18 = 31 \\ c_{34} + g(4, \{2\}) = 12 + 13 = 25 \end{cases} = 25$

$\therefore g(4, \{2,3\}) = \min \begin{cases} c_{42} + g(2, \{3\}) = 8 + 15 = 23 \\ c_{43} + g(3, \{2\}) = 9 + 18 = 27 \end{cases} = 23$

GOOD LUCK

TOPIC NAME: PG: TIME: DATE:

$$g(1,\{2,3,4\}) = \min \begin{cases} c_{12} + g(2,\{3,4\}) = 10+25 = 35 \\ c_{13} + g(3,\{2,4\}) = 15+25 = 40 \\ c_{14} + g(4,\{2,3\}) = 20+23 = 43 \end{cases} = 35$$

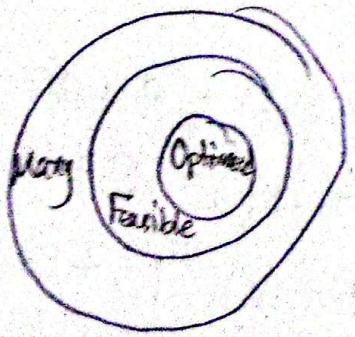
~~Path = 1 → 2 → 4 → 3 → 1~~

∴ Optimal tour length = 35

∴ " " = 1 → 2 → 4 → 3 → 1

Optimization Solution:

* Optimized → minimum or maximum solution subject to constraint



* There can be multiple optimal solutions

GOOD LUCK

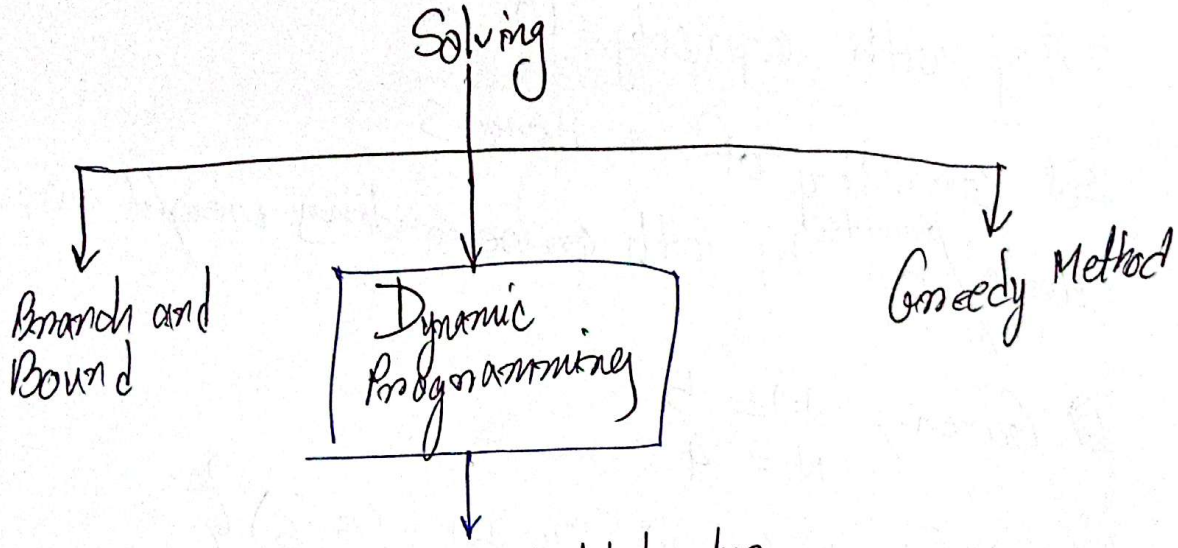
TOPIC NAME : 10 - 9

DAY : C-17

TIME :

DATE : 30 / 1 / 25

Optimization Problem



* Gives a tabulated value
* Can only be done if there are smaller problems
* ~~Dependent~~ Sub Problems
  ↳ ~~Dependent~~ For independent sub problems use Divide and Conquer

Steps for DP:

Think → Math → Table/Store → Result

GOOD LUCK

0-1 Knapsack

Bag with capacity W

Set containing of n items S

Items/Bundles b_i with corresponding weight w_i

Given, $W = 5$

$N = 4$

$S\{(w_1, b_1)\} = \{(2, 3), (3, 4), (4, 5), (5, 6)\}$

i/W	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	3	3+0	3+0	3+0
2	0	0	3	4+0	4+0	4+3 = 7
3	0	0	3	4	5+0	5+0 = 5 X
4	0	0	3	4	5	7

Optimal solution

$$V[k, w] = \begin{cases} V[k-1, w] & \text{if } w_k > w \\ \max\{V[k-1, w], (V[k-1, w-w_k] + b_k)\} & \text{else} \end{cases}$$

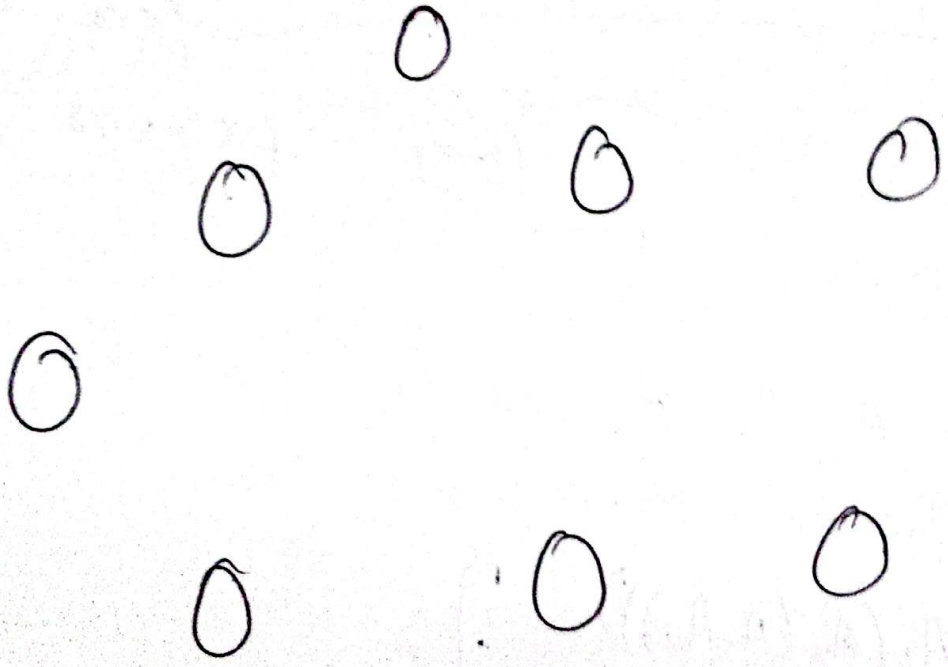
Value with item

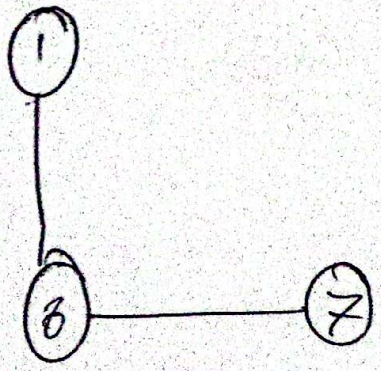
TOPIC NAME: U-9 DAY L-8

TIME: DATE 04/02/25

Lab Final: 26/02/25
8:50

Offline-3: ~~0/1 Knapsack~~ MCM





GOOD LUCK

TOPIC NAME : W-9 DAY : C-18

TIME : DATE : 4/2/25

Matrix Chain Multiplication

Complexity of Matrix Multiplication: $p \times q$ matrix

$q \times r$ "

Comp = $p \times q \times r$

For A_1 A_2 A_3 and A_4,

Multiplication processes are:

1. $(A_1 (A_2 (A_3 A_4)))$
2. $(A_1 (A_2 A_3) A_4)$
3. $(((A_1 A_2) A_3) A_4))$
4. ~~$(A_1 A_2) A_3$~~
4. $(A_1 A_2)(A_3 A_4)$
5. $(A_1 (A_2 A_3) A_4))$

We pick whichever sequence has least cost

GOOD LUCK

TOPIC NAME : DAY : TIME : DATE : / /

For $A_{10\times100}$, $B_{100\times5}$, $C_{5\times50}$

$(AB)C = \{(10\times100\times5) + (10\times5\times50)\}$

$= 50000 + 5000$

$= 5000$

$AB_{10\times5}\ C_{5\times50}$

$= 10\times5\times50$

$= 2500$

$(AB)+((AB)C) = 7500$

$(A(BC)) = BC_{100\times50} = (100\times5\times50)$

$= 25000$

$= A(BC) = 10\times100\times50$

$= 50000$

$(BC)+(A(BC)) = 75000$

GOOD LUCK

TOPIC NAME

Optimal

$((A_1 A_2 \cdots A_k)(A_{k+1} A_{k+2} \cdots A_n))$

$A_{i \cdots j} = A_i A_{i+1} \cdots A_j$

$= (A_i A_{i+1} \cdots A_k)(A_{k+1 \cdots j} A_j)$

$= m[i,k] + m[k+1, j]$

$\rightarrow p_{i-1} \times p_k \times p_j$

$\therefore m[i,j] = m[i,k] + m[k+1, j] + p_{i-1} p_k p_j$

GOOD LUCK

TOPIC NAME: 10-9 DAY: C-19

TIME: DATE: 5/2/25

$$m[i,j] = \begin{cases} 0 & \text{if } i = j \\ \min_{i \le k < j} \{ m[i,k] + m[k+1,j] + p_{i-1} p_k p_j \} & \text{if } i < j \end{cases}$$

$n = 5 \quad p = (10, 5, 1, 10, 2, 10)$

$[10 \times 5]_0 \times [5 \times 1] \times [1 \times 10] \times [10 \times 2] \times [2 \times 10]$

m

	1	2	3	4	5
1	0	50	150	90	190
2		0	50	30	90
3			0	20	40
4				0	200
5					0

s

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

for value of k

GOOD LUCK

TOPIC NAME : DAY :

TIME : DATE : / /

~~m[1,2] =~~

$(A_1)(A_2)$

Here,

For 1 and 2,

$i = 1$
$j = 2$
$k = 1$

$\therefore m[1,2] = m[1,1] + m[2,2] + P_0 P_1 P_2$

$= 0 + 0 + (10 \times 5 \times 1)$

$= 50$

For 3 and 4,

$i = 3$
$j = 4$
$k = 3$

$m[3,4] = m[3,3] + m[4,4] + P_2 P_3 P_4$

$= 0 + 0 + 1 \times 10 \times 2$

$= 20$

For 4 and 5,

$i = 4$
$j = 5$
$k = 4$

$m[4,5] = m[4,4] + m[5,5] + P_3 P_4 P_5$

$= 0 + 0 + 10 \times 2 \times 10$

$= 200$

GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

For A_1 to A_3 and $k = 1$

$i = 1$
$j = 3$
$k = 1$

$$m[1,3] = m[1,1] + m[2,3] + P_0 P_1 P_3$$

$$= 0 + 50 + (10 \times 5 \times 10)$$

$$= 50 + 500$$

$$= 550$$

$A_2(A_3 A_4)$

For A_2 to A_4 and $k = 3$

$i = 2$
$j = 4$
$k = 3$

$$m[2,4] = m[2,3] + m[4,4] + P_1 P_3 P_4$$

$$= 50 + 0 + 5 \times 10 \times 2$$

$$= 50 + 100$$

$$= 150$$

GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

For A_1 to A_4 and $k=2$

$i=1$
$j=4$
$k=2$

$$m[1,4] = m[1,2] + m[3,4] + P_0 \times P_2 \times P_4$$

$$= 50 + 20 + 10 \times 1 \times 2$$

$$= 70 + 20$$

$$= 90$$

For A_1 to A_5 and $k=2$

$i=1$
$j=5$
$k=2$

$$m[1,5] = m[1,2] + m[3,5] + P_0 P_2 P_5$$

$$= 50 + 40 + 10 \times 1 \times 10$$

$$= 190$$

GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

	1	2	3	4	5
1	0	50	150	90	190
2		0	50	30	90
3			0	20	40
4				0	200
5					0

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

∴ Required Answer:

$$\left((A_1)(A_2)\right)\left(\left((A_3)(A_4)\right)(A_5)\right) : 190$$

GOOD LUCK

TOPIC NAME: W-9 DAY: C-20

TIME: DATE: 6/2/24

Dijkstra(G, ws)	Array	Heap
1.	V	V
2.	1	1
3.	V	VlgV → book says it becomes faster
4.	V	V+1
	$n \cdot V = V^2$	VlgV
5.	V	
6.		V
7.	E	E
8.	E	ElgV
	$V^2 + E = V^2$	$VlgV + ElgV$ $= (V+E)lgV \approx ElgV$

GOOD LUCK

TOPIC NAME : W-10 DAY : C-21

TIME : DATE : 11/2/25

~~Subseq~~ LCS

Subsequence: L to R any number of sequential chars.

" Substrings are also subsequence, but a subsequence may not be a substring" — Justify.

* Used in genome sequencing

GOOD LUCK™

TOPIC NAME : DAY :
TIME : DATE : / /

$X = \langle x_1, x_2 \ldots x_m \rangle$

$Y = \langle y_1, y_2 \ldots y_n \rangle$

X_i = ith prefix of $X \rightarrow \langle x_1, x_2 \ldots x_i \rangle$

Y_j = jth " " $Y \rightarrow \langle y_1, y_2 \ldots y_j \rangle$

$$c[i,j] = \begin{cases} 0 & \text{if } i=0 \text{ or } j=0 \\ c[i-1, j-1] + 1 & \text{if } i,j > 0 \text{ and } x_i = y_j \\ \max(c[i, j-1], c[i-1, j]) & \text{if } i,j > 0 \text{ and } x_i \neq y_j \end{cases}$$

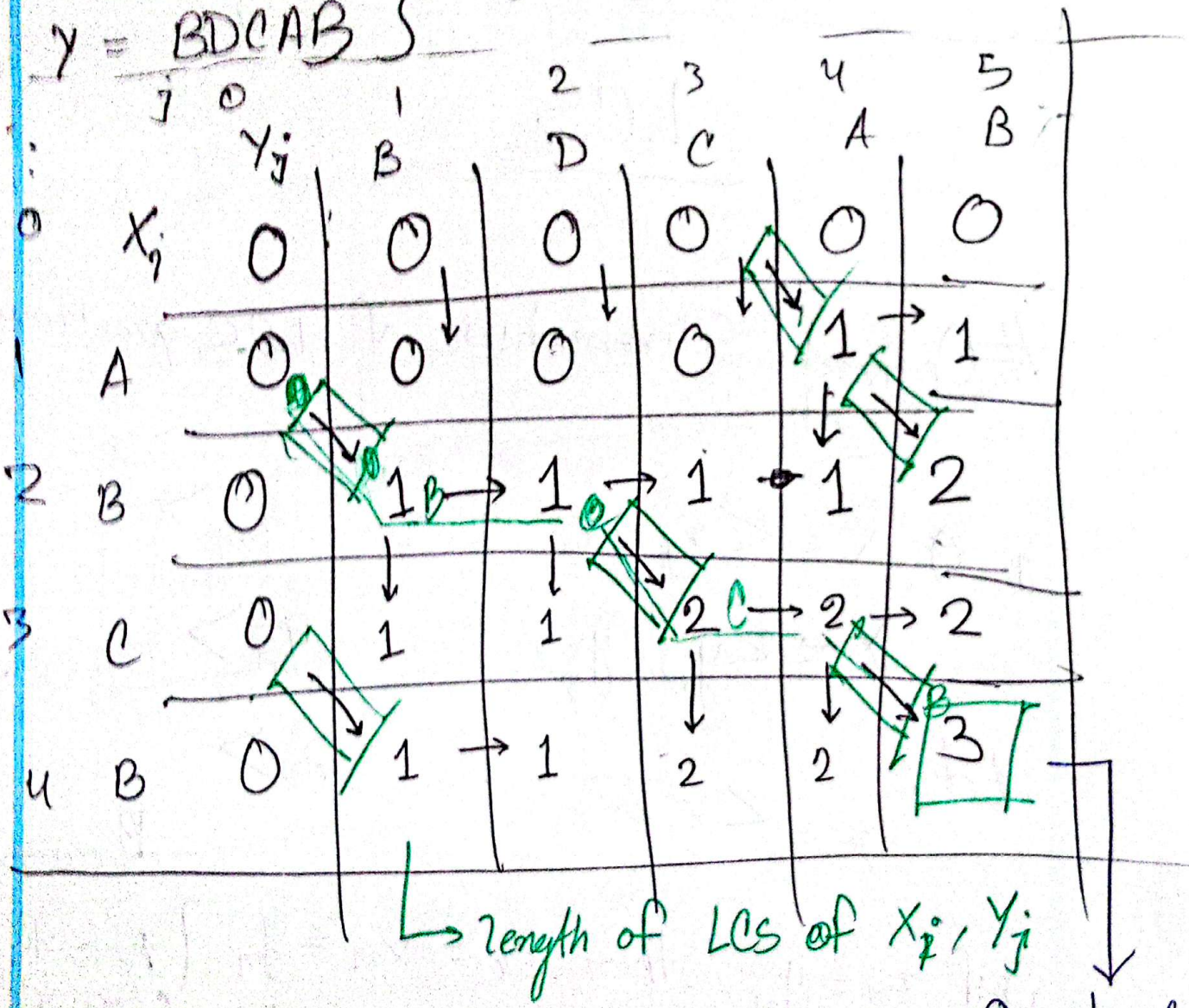
$b[i,j]$ = arrow matrix

↳ you can use pairs

GOOD LUCK

TOPIC NAME: DAY:
TIME: DATE: / /

X = ABCB
Y = BDCAB
} Find LCS



* Check ~~$X_{i-1} \neq Y_j$~~ dont ↑ and ← to verify if they match
* If (match) → ↖ + 1

Ans: BCB : 3

GOOD LUCK

TOPIC NAME: W-10 DAY: C-23(22-Quiz)

TIME: DATE: 13/2/25

Quiz-4 + (Lab Final): DP ~~ation~~ Programs

LCS

#Optimal Substructure of LCS questions

a. $X = \langle x_1, x_2 \ldots x_m \rangle$

$Y = \langle y_1, y_2 \ldots y_n \rangle$

Let, $Z = \langle z_1, z_2 \ldots z_k \rangle$

Reason

i) if $x_m = y_n$, then $z_k = x_m = y_n$ and Z_{k-1} is LCS of X_{m-1} and Y_{n-1} } ~~Adds 1~~ for adding 1

ii) If $x_m \neq y_n$, then $z_k \neq x_m$ implies that Z is an LCS of X_{m-1} and Y

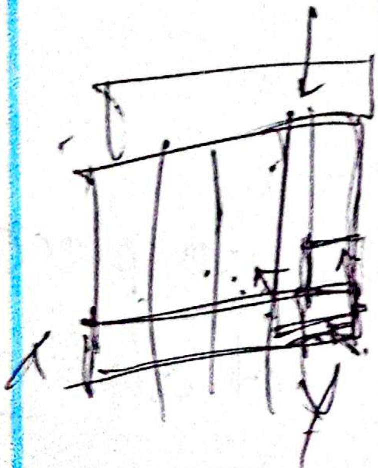
iii) If $x_m \neq y_n$ and $z_k \neq y_n$ implies that Z is an LCS X and Y_{n-1} } max (left, up)

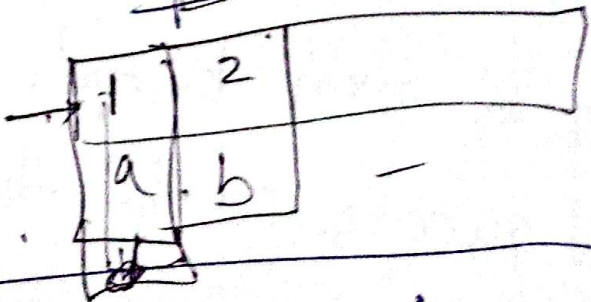
GOOD LUCK

TOPIC NAME : W-11 DAY: L-11? TIME: DATE: 18/2/25

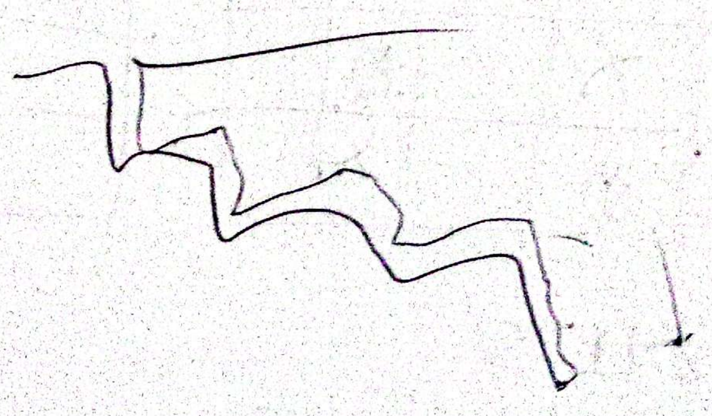
Online: CD A B A E
AB C D E B A F

if common ~~with~~
- ~~push~~ if none → push while its last sdx ≤ com
- if ~~data.2~~ next elsewhere > last data of any ≤ common → push
- else → push elsewhere
  - skip





Online-8: Sum of Subset, N-Queens



TOPIC NAME: W-11 DAY: C-24

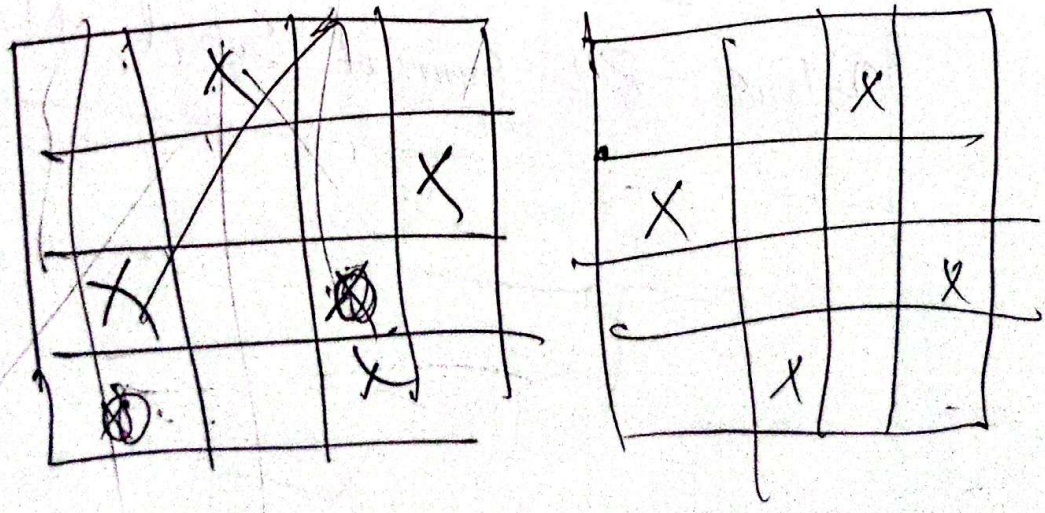
TIME: DATE: 18/2/25

Back-tracking

* We go back a node to find the next solution.

N-Queen

in an nxn board, we have to place N queens so that they are not attacking



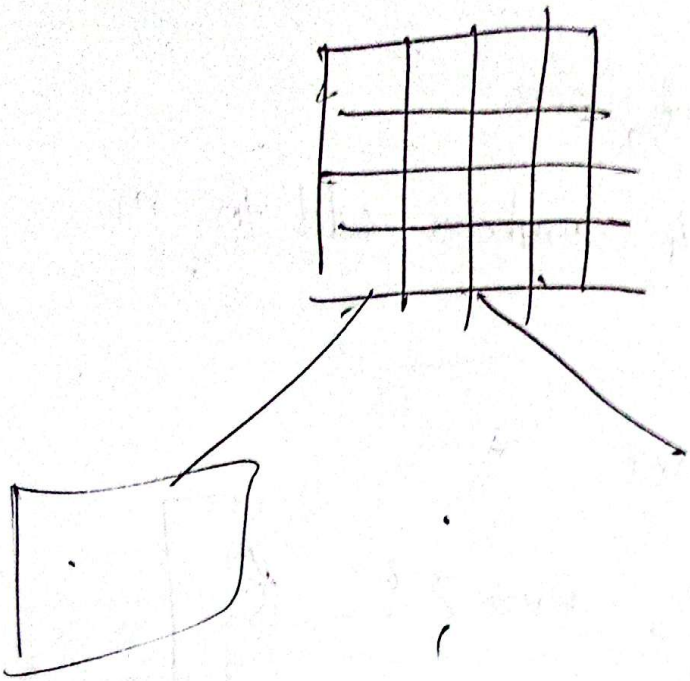
Output: 2 4 1 3

↳ columns to place queens

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

How to use :



GOOD LUCK

TOPIC NAME : 10-11 DAY : C-26 (25 was Taki)

TIME : DATE : 20/2/25

Sum of Subsets

~~Sum~~ in a set, ~~the~~

find which numbers add to m

input : m , set

output : { num1, num2 ... }

{ 0 ind1, ind2 . . . } → Either one

{ 0 for not 1 for taken }

checks : ~~if sorted~~ if starting sum > m → stop

< m → "

GOOD LUCK

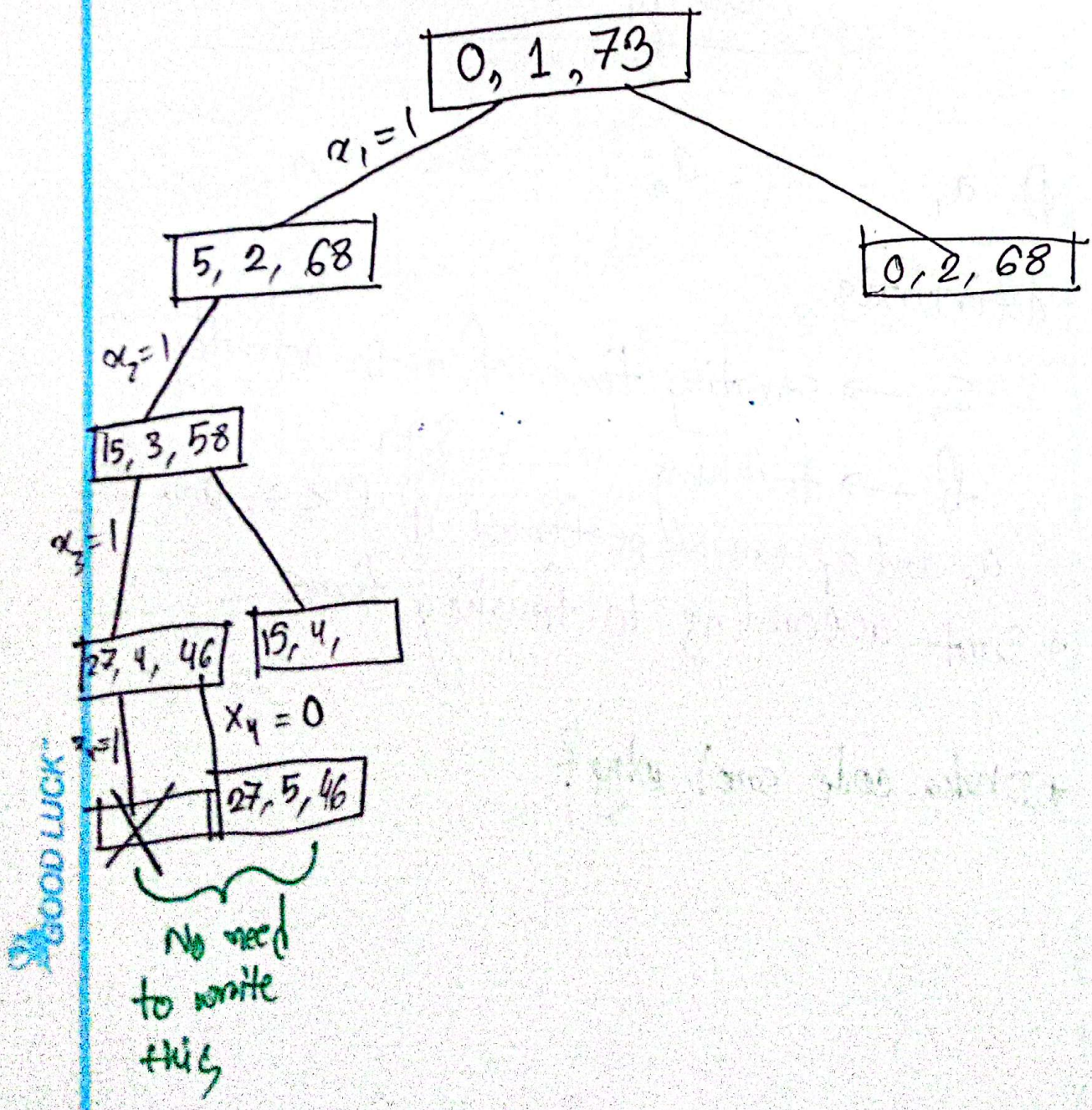
TOPIC NAME _____ DAY : _____

TIME : DATE : / /

$n = 6$ $w = 30$

$w[1:6] = \{5, 10, 12, 13, 15, 18\}$

curr_sum, next_elem (index), sum_of_remaining



"GOOD LUCK"

TOPIC NAME : W-12 DAY : C-28

TIME : DATE : 26 / 2 / 25

Greedy Method

* Whatever is best at the moment

Activity Selection Problem

① $a_1 - - - - a_i - - - - a_n$

activitities

$s_i \rightarrow$ starting time of i-th activity

$f_i \rightarrow$ finishing " " i-th "

a_i and a_j can be performed if $f_i \leq s_j$ or $f_j \leq s_i$

* Sort according to finishing time

* Dhaka code smol why?

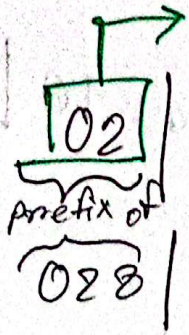
GOOD LUCK

TOPIC NAME : W-12 DAY : C-29

TIME : DATE : 27/2/25

Huffman

prefix code



8331150

8331150

Use smaller codes for the frequent usage

No string is a prefix of another.

For variable table in slide

$45 \times 1 + 13 \times 3 + 12 \times 3 + 16 \times 3 + 9 \times 4 + 5 \times 4$

$= 224$

$= 224$ byte

69

~~+ 4 4~~

~~3 6 8~~

290

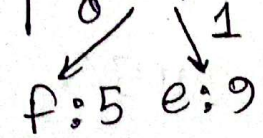
GOOD LUCK

TOPIC NAME : DAY :

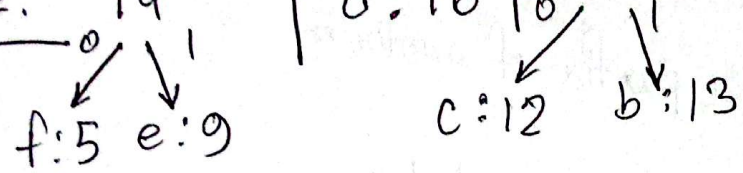
TIME : DATE : / /

Start: f:5 | e:9 | c:12 | b:13 | d:16 | a:45

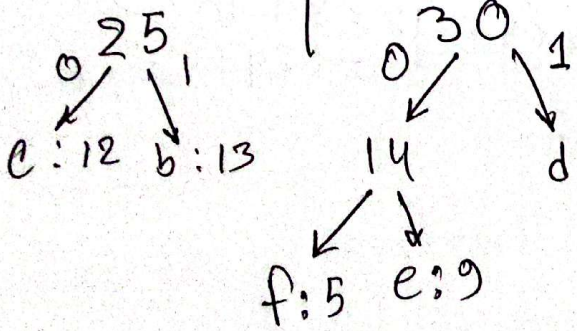
Step 1: c:12 | b:13 | 14 | d:16 | a:45



Step 2: 14 | d:16 | 25 | a:45



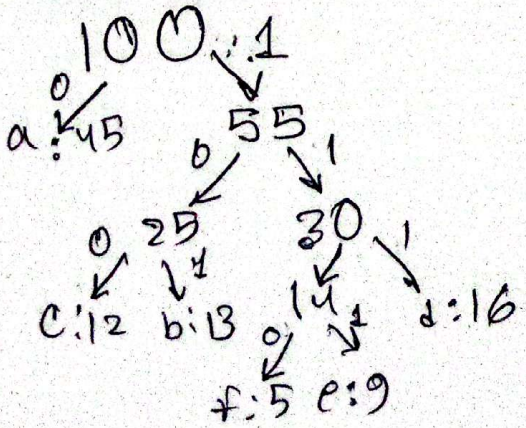
Step 3: 25 | 30 | a:45



Step 4:

:

Step 5:



Here

a = 0

b = 101

c = 100

d = 111

e = 1101

f = 1100

GOOD LUCK

TOPIC NAME: W-13 DAY: C-29

TIME: DATE: 4/3/25

Fractional Knapsack

Items:	1	2	3
w_i	18	15	10
p_i	25	24	15
Value: (p_i/w_i)	1.39	1.6	1.5

∴ We take:

15 of i_2

5 of i_3

0 of i_1

~~Proofi~~

Solution:

$(x_1, x_2, x_3) = (0, 1, 1/2)$

Profit $= (25*0) + (24*1) + (15*1/2)$

$= 31.5$

*Sort using unit price

GOOD LUCK

TOPIC NAME: 10 - 13

C - 21

Date: 6 / 3 / 25

NP - Completeness

*Time for Research

Polynomial

Linear Search — n

Binary " — $\log n$

Bubble Sort — n^2

Merge sort — $n \log n$

Matrix Multiplication — n^3

Exponential

0/1 Knapsack — 2^n

TSP — 2^n

Sum of Subset — 2^n

Graph Coloring — 2^n

Polynomial Time:

If complexity of any algorithm can be expressed in polynomial time then it is in polynomial time.

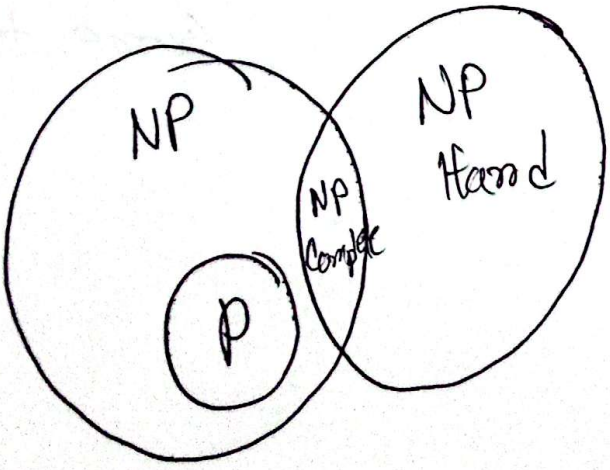
GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

Algorithm

- **Class P**
  - all lines are deterministic and in polynomial time
- **Class NP**
  - all lines may not be deterministic but in polynomial time
- **Class NP hard**
- **NP Complete**
  - if an NP-hard problem can have a non-deterministic solution, it is NP Complete



GOOD LUCK

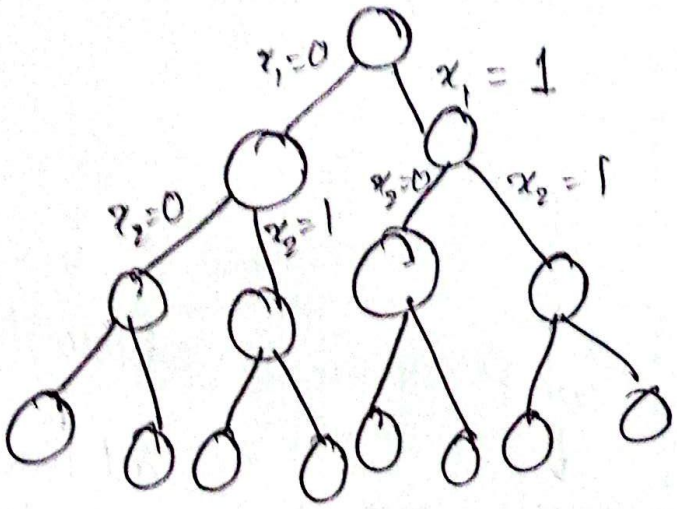
TOPIC NAME

DAY

TIME : DATE : / /

For Satisfiability Problem:

x_1	x_2	x_3	F
0	0	0	
0	0	1	
.			
.			
.			
1	1	1	



For ~~Satisfiabil~~ 0/1 Knapsack:

I_1	I_2	I_3	W
0	0	0	
0	0	1	
.			
.			
.			
1	1	1	

Same tree

Check Common

GOOD LUCK