DAA

UNIT 1 **Introduction to Algorithms** **9 hours**

Introduction to Algorithms: Algorithm specification, Performance Analysis. Divide and Conquer: The general method: Binary search, finding maximum and minimum, Merge sort, Quick sort, Selection, Strassen's Matrix multiplication.

UNIT 2 **The Greedy Method** **9 hours**

The Greedy Method: The general method, Knapsack problem, Job sequencing with deadlines, optimal storage on tapes, minimum cost spanning trees, single source shortest paths.

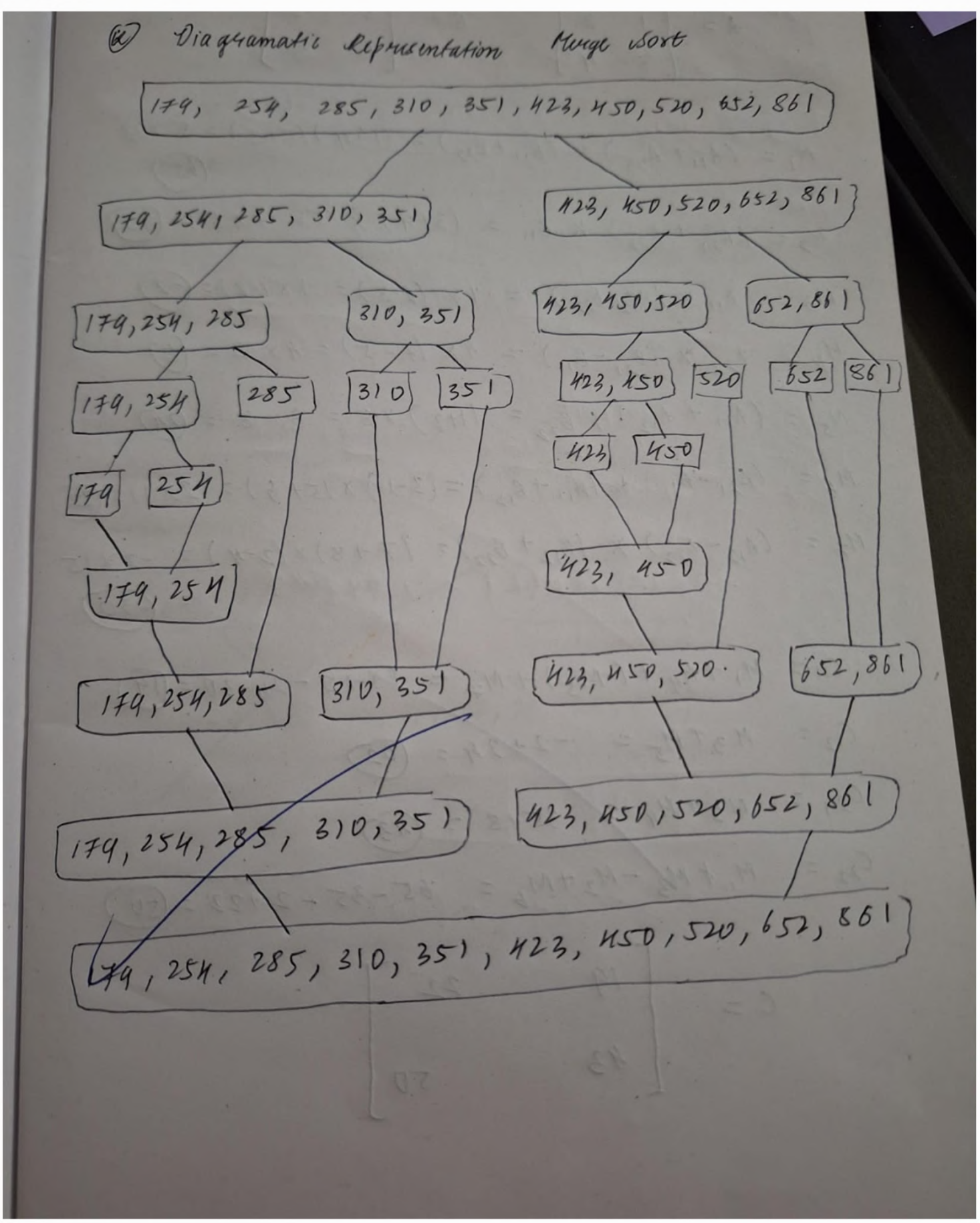
UNIT 3 **Dynamic Programming** **9 hours**

Dynamic Programming: The general method, multistage graphs, all pairs shortest paths, optimal binary search trees, reliability design, the travelling sales person problem.

UNIT - 1

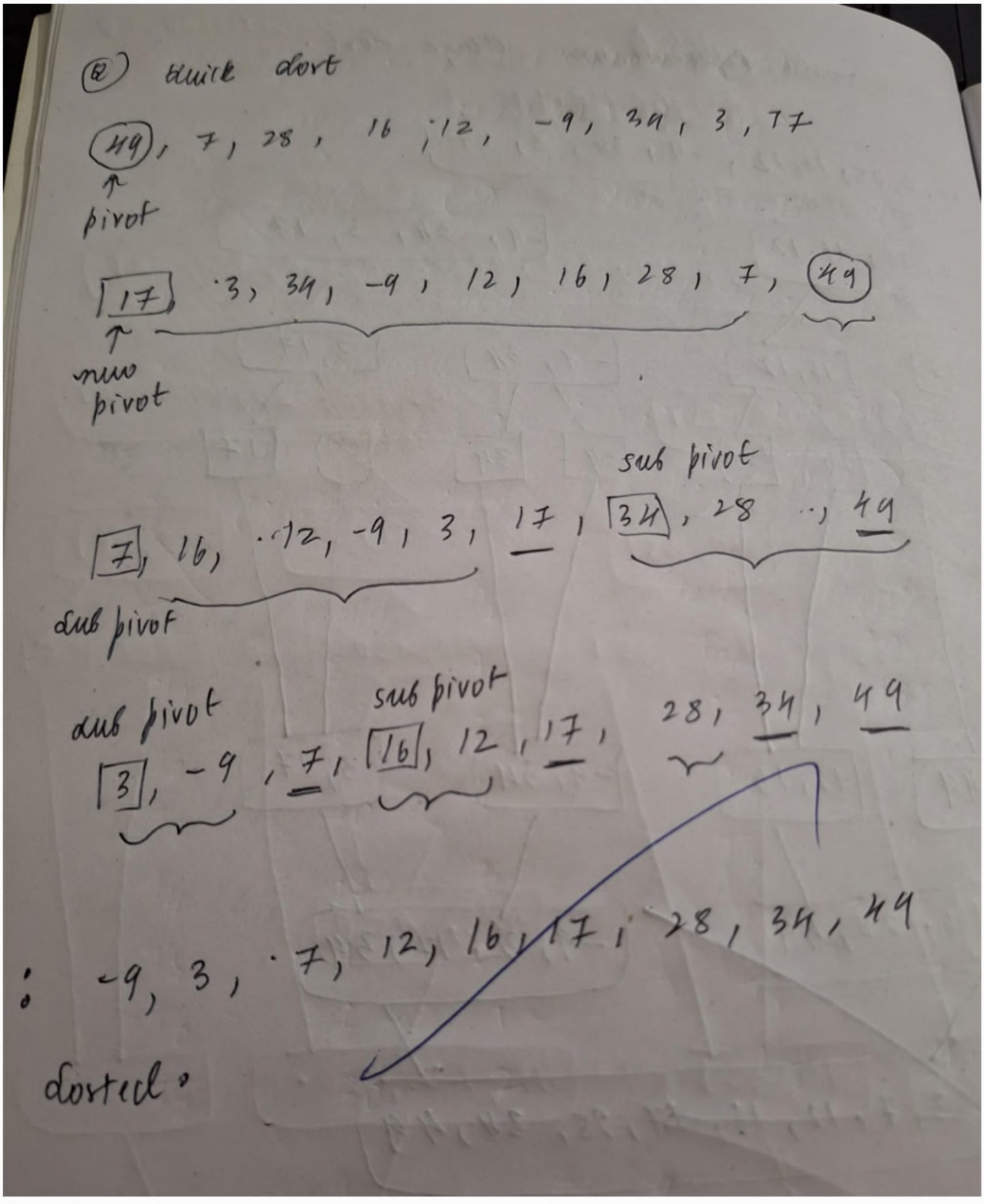
- MERGE SORT

Best, Avg, Worst cases → O(n log n)



= QUICK SORT

Best, Avg cases → O(n log n)
Worst case → O(n²)



- BINARY SEARCH

Divides from middle and searches

Best case → O(1) Avg and Worst cases → O(log n)
(Successful search)

All cases → O(log n) (Unsuccessful search)

– FINDING MIN & MAX

6. Finding maximum and minimum

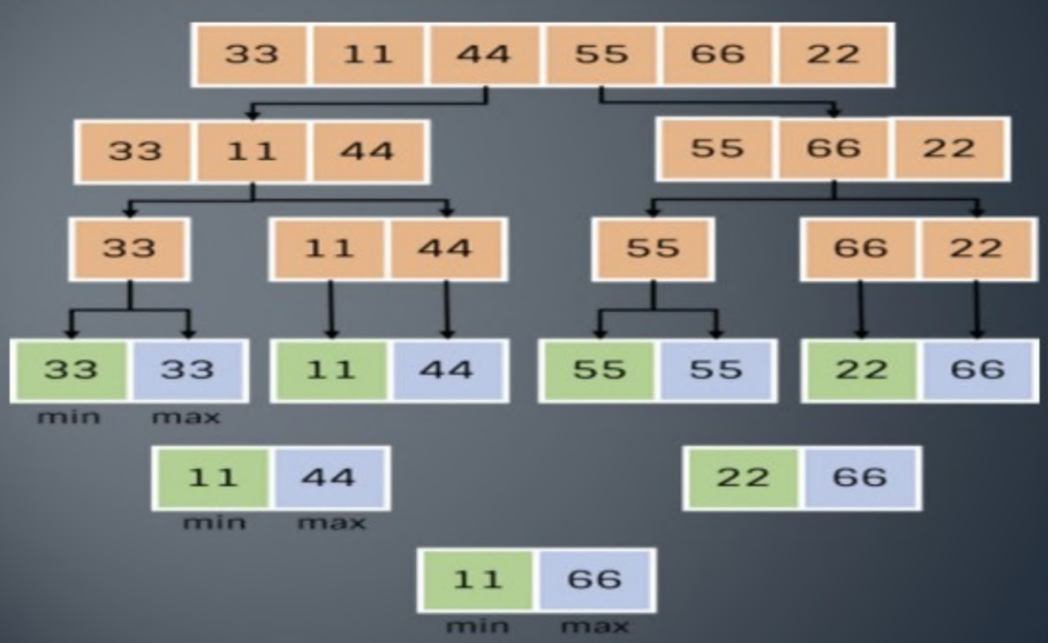
- Divide and conquer approach for Max. & Min problem works in three stages.

  Stage1: If a_1 is the only element in the array, a_1 is the maximum and minimum.

  Satge2: If the array contains only two elements a_1 and a_2, then the single comparison between two elements can decide the minimum and maximum of them.

  Stage3: If there are more than two elements, the algorithm divides the array from the middle and creates two sub-problems. Both sub-problems are treated as an independent problem and the same recursive process is applied to them. This division continues until sub-problem size becomes one or two.

For example, **find max and min from the sequence <33, 11, 44, 55, 66, 22> using divide and conquer approach**



— STRASSEN'S MATRIX MULTIPLICATION

(8) Strassen's algorithm 2×2 Matrix Multiplication

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \qquad B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$M_1 = (A_{11} + A_{22}) * (B_{11} + B_{22}) = (1+4)(5+8) = 5 \times 13 = 65$

$M_2 = (A_{21} + A_{22}) * B_{11} = (3+4) \times 5 = 7 \times 5 = 35$

$M_3 = A_{11} * (B_{12} - B_{22}) = 1 \times (6-8) = 1 \times (-2) = -2$

$M_4 = A_{22} * (B_{21} - B_{11}) = 4 \times (7-5) = 4 \times 2 = 8$

$M_5 = (A_{11} + A_{12}) * B_{22} = (1+2) \times 8 = 3 \times 8 = 24$

$M_6 = (A_{21} - A_{11}) * (B_{11} + B_{12}) = (3-1) \times (5+6) = 2 \times 11 = 22$

$M_7 = (A_{12} - A_{22}) * (B_{12} + B_{22}) = (7+8) \times (2-4) = -2 \times 15 = -30$

$C_{11} = M_1 + M_4 + (-M_5) + M_7 = 65 + 8 - 24 - 30 = 19$

$C_{12} = M_3 + M_5 = -2 + 24 = 22$

$C_{21} = M_2 + M_4 = 35 + 8 = 43$

$C_{22} = M_1 + M_3 - M_2 + M_6 = 65 - 35 - 2 + 22 = 50$

$$C = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

UNIT-2

— KNAPSACK PROBLEM

(12) Knapsack problem

$n = 7$, $m = 15$ $(p_1, p_2, \dots p_7) = (10, 5, 15, 7, 6, 18, 3)$

$(w_1, w_2, \dots w_7) = (2, 3, 5, 7, 1, 4, 1)$

$n \rightarrow$ no. of items $m \rightarrow$ max weight

$p \rightarrow$ profits $w \rightarrow$ weights

Calculating profit-to-weight ratio $\left(\frac{p_i}{w_i}\right)$

$R_1 = \left(\frac{10}{2}\right) = 5$ $R_5 = \left(\frac{6}{1}\right) = 6$

$R_2 = \left(\frac{5}{3}\right) \simeq 1.67$ $R_6 = \left(\frac{18}{4}\right) = 4.5$

$R_3 = \left(\frac{15}{5}\right) = 3$ $R_7 = \left(\frac{3}{1}\right) = 3$

$R_4 = \left(\frac{7}{7}\right) = 1$

Sorting based on ratio.

$R_5, R_1, R_6, R_3, R_7, R_2, R_4$

Adding items to knapsack based on sorted order

(1) Item 5 : Profit 6, Weight 1 Remaining capacity = 14

(2) I_1 : $p :- 10$, $w :- 2$, $RC = 12$

(3) I_6 : $p :- 18$, $w :- 4$, $RC = 8$

(4) I_3 : $p :- 15$, $w :- 5$, $RC = 3$

(5) I_7 : $p :- 3$, $w :- 1$, $RC = 2$

There is no remaining capacity for adding items.

Total profit : $6+10+18+15+3 = (52)$

total weight : $1+2+4+5+1 = (13)$

Max profit : (52) Min weight : (13)

– JOB SEQUENCING WITH DEADLINES

(a) What is the solution generated by the job sequence with deadlines

$m = 7 \quad (p_1, p_2, p_3 \cdots p_7) = (3, 5, 20, 18, 1, 6, 30)$

$(d_1, d_2, \cdots d_7) = (1, 3, 4, 3, 2, 1, 2)$

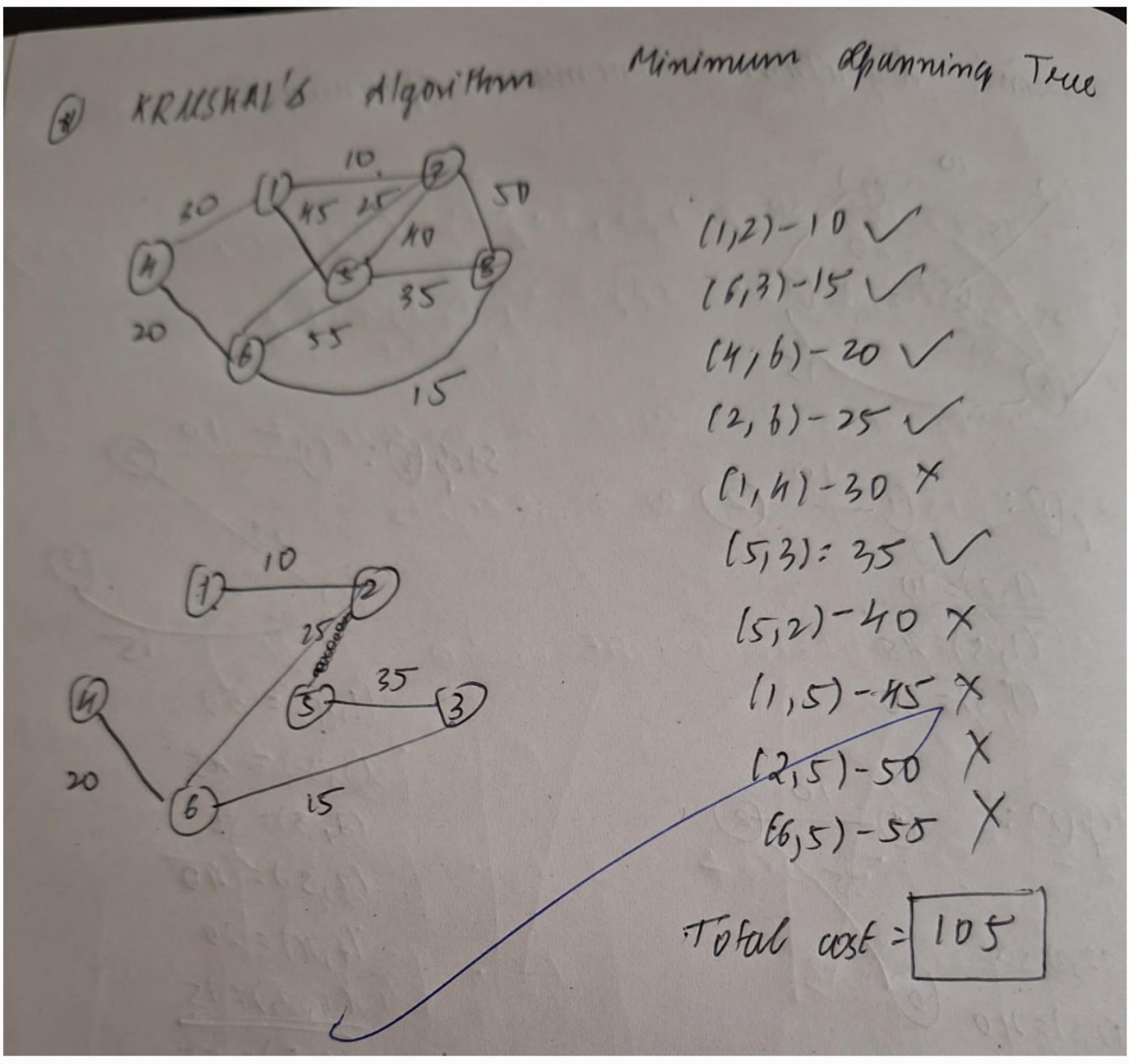
Task	Profit	D_L
T_1	3	1
T_2	5	3
T_3	20	4
T_4	18	3
T_5	1	2
T_6	6	1
T_7	30	2

task	Profit	D_L
T_7	30	2 ✓
T_3	20	4 ✓
T_4	18	3 ✓
T_6	6	1 ✓
T_2	5	3 ✗
T_1	3	1 ✗
T_5	1	2 ✗

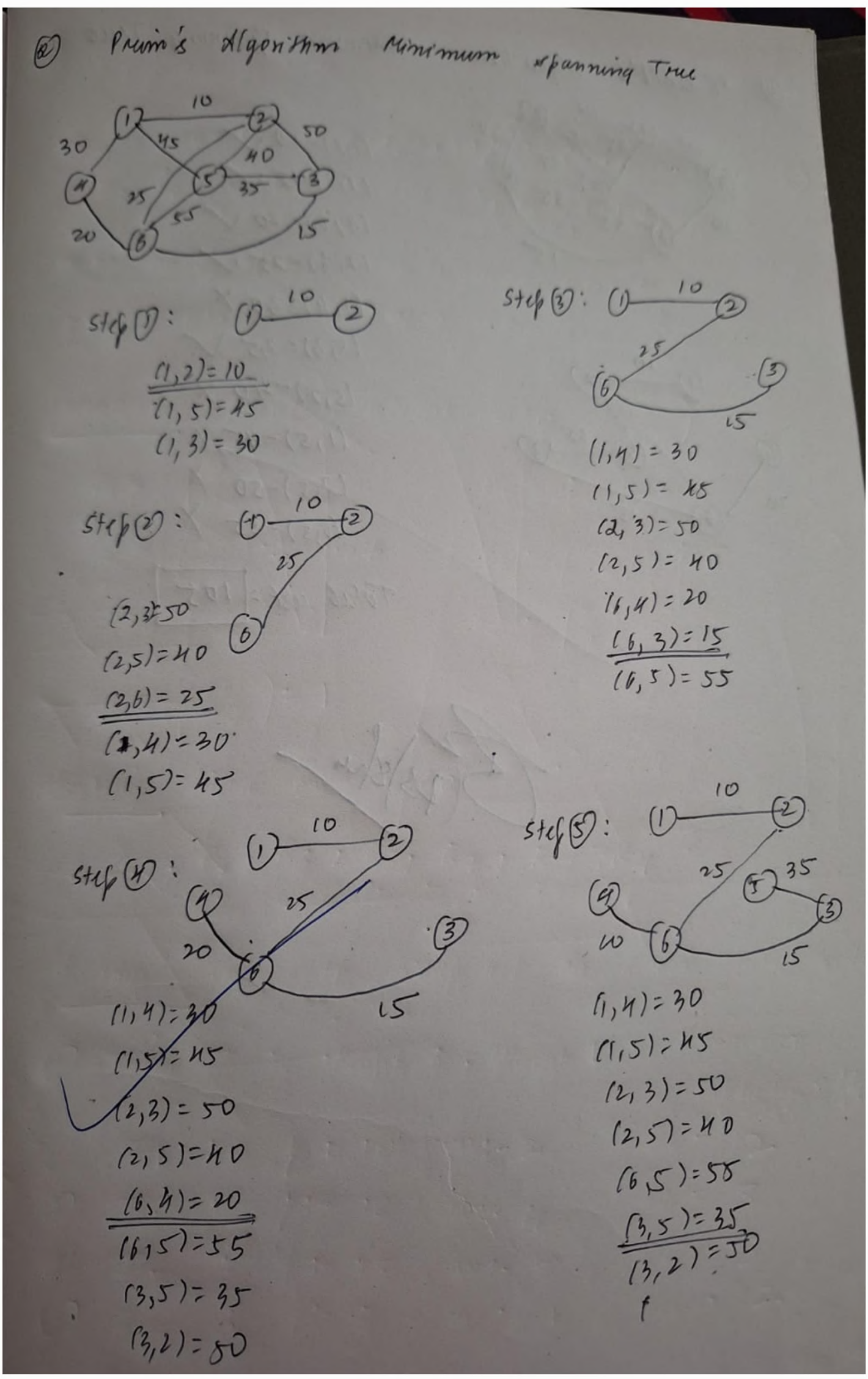
1	2	3	4
6	30	18	20

$T_6 + T_7 + T_4 + T_3 = (74)$

– KRUSKAL'S ALGORITHM



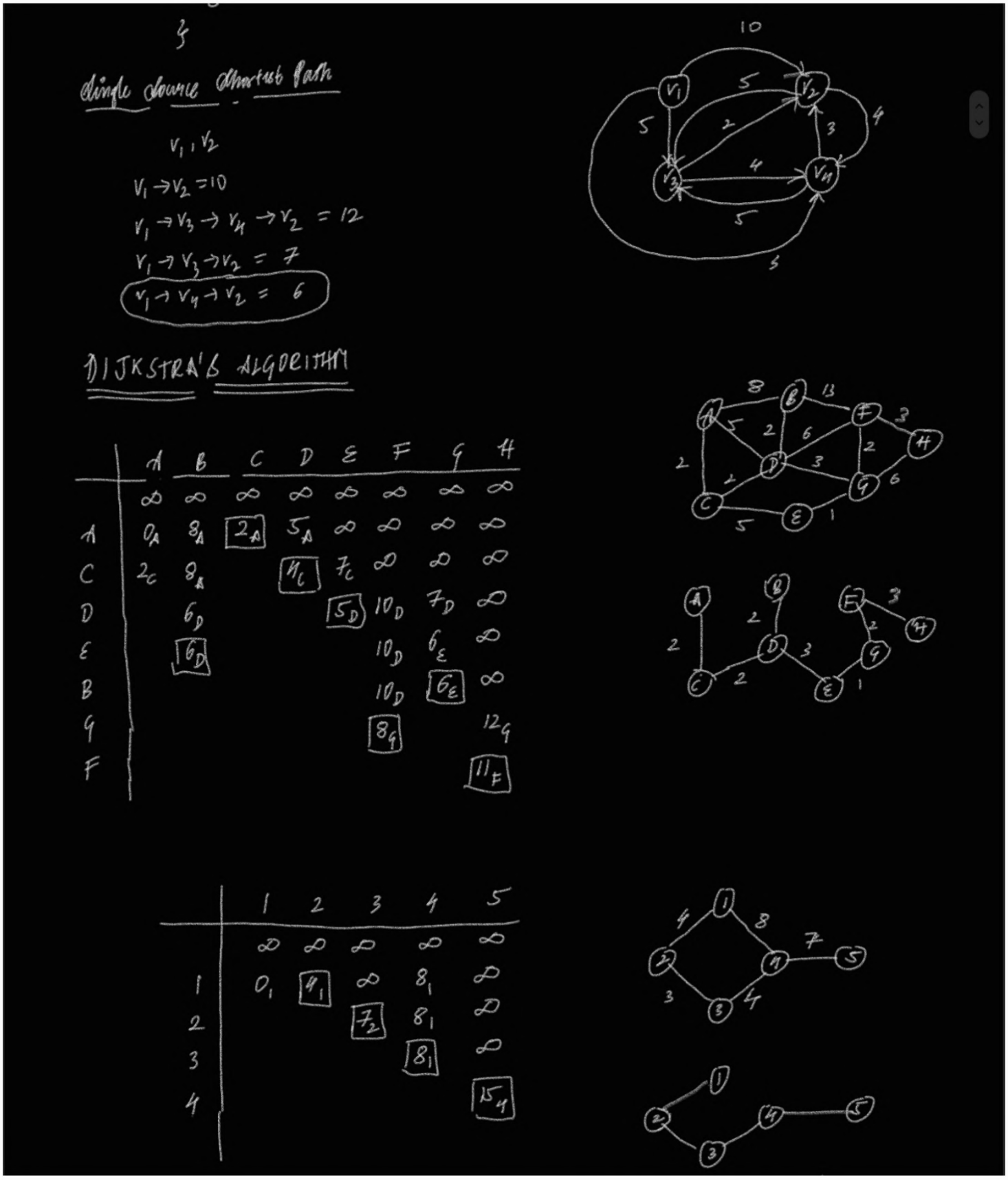
– PRIM'S ALGORITHM



- SINGLE SOURCE SHORTEST PATHS

(OR)

DIJKSTRA'S ALGO



- OPTIMAL STORAGE OF TAPES

For e.g. Suppose there are 3 programs of lengths 2, 5 and 4 respectively. So there are total 3! = 6 possible orders of storage.

	ORDER	TOTAL RETRIEVAL TIME	MEAN RETRIEVAL TIME
1	1 2 3	2 + (2 + 5) + (2 + 5 + 4) = 20	20/3
2	1 3 2	2 + (2 + 4) + (2 + 4 + 5) = 19	19/3
3	2 1 3	5 + (5 + 2) + (5 + 2 + 4) = 23	23/3
4	2 3 1	5 + (5 + 4) + (5 + 4 + 2) = 25	25/3
5	3 1 2	4 + (4 + 2) + (4 + 2 + 5) = 21	21/3
6	3 2 1	4 + (4 + 5) + (4 + 5 + 2) = 24	24/3

It's clear that by following the second order in storing the programs, the mean retrieval time is least.

UNIT-3

– ALL PAIR SHORTEST PATH

All Pairs shortest Path

$A^k(i,j)$ shortest path b/w vertices i & j where k is the intermediate node.

$$A^k(i,j) = \min\left[A^{k-1}(i,j),\ A^{k-1}(i,k) + A^{k-1}(k,j)\right]$$

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 4 & 15 \\ 8 & 0 & 2 \\ 3 & \infty & 0 \end{bmatrix} \end{array}$$

Let $k = 1$

$A^1(1,2) = \min\left[A^{1-1}(1,2),\ A^{1-1}(1,1) + A^{1+1}(1,2)\right]$

$= \min[4,\ 0+4] = 4$

$A^1(1,3) = \min\left[A^0(1,3),\ A^0(1,1) + A^0(1,3)\right] = \min[15,\ 0+15] = 15$

$A^1(2,1) = \min\left[A^0(2,1),\ A^0(2,1) + A^0(1,1)\right] = \min[8,\ 8+0] = 8$

$A^1(2,3) = \min\left[A^0(2,3),\ A^0(2,1) + A^0(1,3)\right] = \min[2,\ 8+15] = 2$

$A^1(3,1) = \min\left[A^0(3,1),\ A^0(3,1) + A^0(1,1)\right] = \min[3,\ 3+0] = 3$

$A^1(3,2) = \min\left[A^0(3,2),\ A^0(3,1) + A^0(1,2)\right] = \min[\infty,\ 3+4] = 7$

for $k=1 \longrightarrow$

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 4 & 15 \\ 8 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix} \end{array}$$

Let $k=2$

$A^2(1,2) = \min\left[A^{2-1}(1,2),\ A^{2-1}(1,2) + A^{2-1}(2,2)\right] = \min[4,\ 4+0] = 4$

$A^2(1,3) = \min\left[A^1(1,3),\ A^1(1,2) + A^1(2,3)\right] = \min[15,\ 4+2] = 6$

$A^2(2,1) = \min\left[A^1(2,1),\ A^1(2,2) + A^1(2,1)\right] = \min[8,\ 0+8] = 8$

$A^2(2,3) = \min\left[A^1(2,3),\ A^1(2,2) + A^1(2,3)\right] = \min[2,\ 0+2] = 2$

$A^2(3,1) = \min\left[A^1(3,1),\ A^1(3,2) + A^1(2,1)\right] = \min[3,\ 7+8] = 3$

$A^2(3,2) = \min\left[A^1(3,2),\ A^1(3,2) + A^1(2,2)\right] = \min[7,\ 7+0] = 7$

for $k=2 \longrightarrow$

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 4 & 6 \\ 8 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix} \end{array}$$

$A^3(1,2) = \min\left[A^2(1,2),\ A^2(1,3) + A^2(3,2)\right] = \min[4,\ 6+7] = 4$

$A^3(1,3) = \min\left[A^2(1,3),\ A^2(1,3) + A^2(3,3)\right] = \min[6,\ 6+0] = 6$

$A^3(2,3) = \min\left[A^2(2,3),\ A^2(2,3) + A^2(3,2)\right] = \min[2,\ 2+7] = 2$

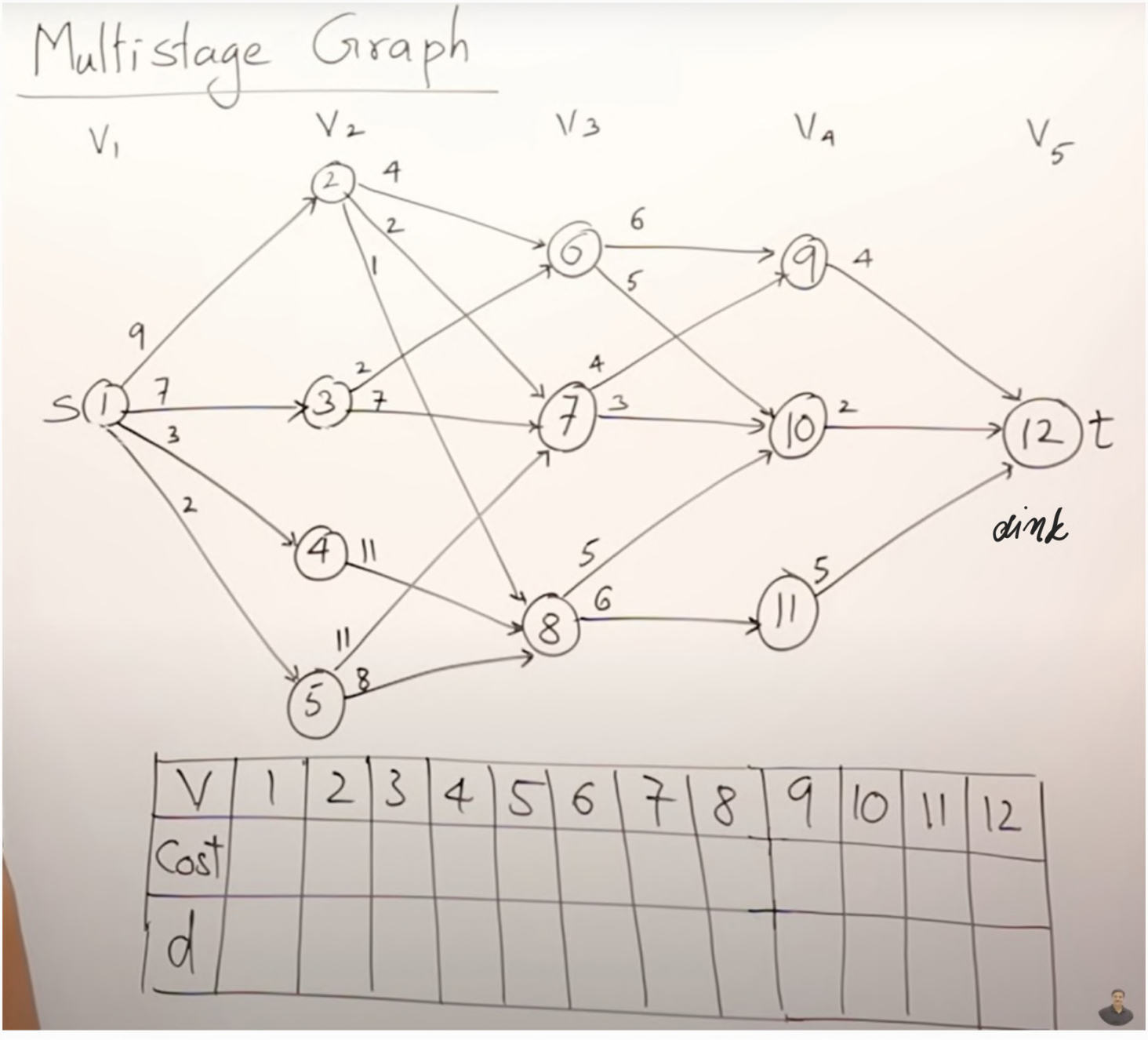
$A^3(3,1) = \min\left[A^2(3,1),\ A^2(3,3) + A^2(3,1)\right] = \min[3,\ 0+3] = 3$

$A^3(3,2) = \min\left[A^2(3,2),\ A^2(3,3) + A^2(3,2)\right] = \min[7,\ 0+7] = 7$

for $k=3 \longrightarrow$

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 4 & 6 \\ 5 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix} \end{array}$$

- MULTI STAGE GRAPH
  - FORWARD METHOD
  - BACKWARD METHOD



V	1	2	3	4	5	6	7	8	9	10	11	12
Cost	16	7	9	18	15	7	5	7	4	2	5	0
d	2/3	7	6	8	8	10	10	10	12	12	12	12

stage (5) $cost(5,12) = 0$

Stage → Vertex

stage (4)

$cost(4,9) = 4$ $cost(4,10) = 2$ $cost(4,11) = 5$

Stage (3)

$$cost(3,6) = \min \begin{cases} c(6,9) + cost(4,9), \\ c(6,10) + cost(4,10) \end{cases}$$

$$= \min \{ 6+4, \underline{5+2} \} = (7)$$

$$cost(3,7) = \min \begin{cases} c(7,9) + cost(4,9), \\ c(7,10) + cost(4,10) \end{cases}$$

$$= \min \{ 4+4, \underline{3+2} \} = (5)$$

$$cost(3,8) = \min \begin{cases} c(8,10) + cost(4,10) \\ c(8,11) + cost(4,11) \end{cases}$$

$$= \min \{ \underline{5+2}, 6+5 \} = (7)$$

Stage (2)

$$cost(2,2) = \min \begin{cases} c(2,6) + cost(3,6), \\ c(2,7) + cost(3,7), \\ c(2,8) + cost(3,8) \end{cases}$$

$$= \min \{ 4+7, \underline{2+5}, 1+7 \} = (7)$$

$$cost(2,4) = \min \{ c(4,8) + cost(3,8)$$

$= 11+7 = (18)$

$$cost(2,5) = \min \begin{cases} c(5,7) + cost(3,7) \\ c(5,8) + cost(3,8) \end{cases}$$

$$= \min \{ 11+5, \underline{8+7} \} = (15)$$

Stage (1)

$$cost(1,1) = \min \begin{cases} c(1,2) + cost(2,2), \\ c(1,3) + cost(2,3), \\ c(1,4) + cost(2,4) \\ c(1,5) + cost(2,5) \end{cases}$$

$$= \min \{ \underset{16}{\underline{9+7}}, \underset{16}{\underline{7+9}}, \underset{21}{\underline{3+18}}, \underset{17}{\underline{2+15}} \}$$

Formula

$$cost(i,j) = \min \{ c(j,l) + cost(i+1, l) \}$$

$d(1,1) = 2$ $\quad d(2,2) = 7 \quad d(3,7) = 10$

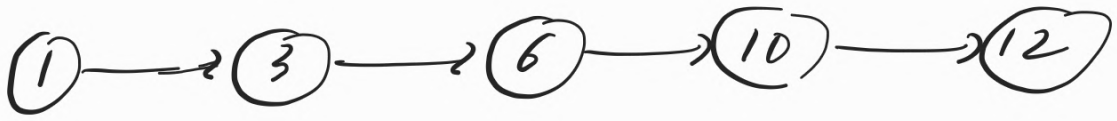
↙ ↳

stage vertex $\quad d(4,10) = 12$

As, $d = 2$ (or) 3 for $d=2$

similarly, we can solve for $d=3$ also

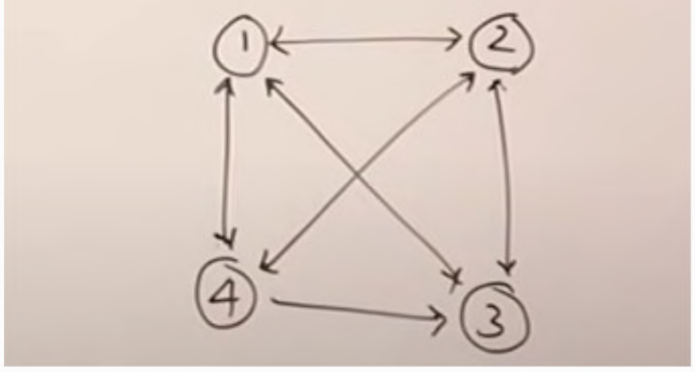
$d(1,1) = 3 \quad d(2,3) = 6 \quad d(3,6) = 10 \quad d(4,10) = 12$



Similarly, backward method which starts from stage ① and goes →

— TRAVELLING SALESMAN PROBLEM

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



$g(2,\phi) = 5$

$g(3,\phi) = 6$

$g(4,\phi) = 8$

$$g(i,S) = \min_{k \in S} \{ C_{ik} + g(k, S-\{k\}) \}$$

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

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

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

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

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

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

26-08-21

TRAVELLING SALESPERSON'S PROBLEM

We can place three cities at 5 from starting vertex 1. The cheat vertex as he can visit either two or three or four. If he visits vertex two after some, he can either 3 or 4.

S=2

$g(2,\{3,4\}) = \min_{j \in \{3,4\}} \{ C_{23} + g(3,4), C_{24} + g(4,3) \} = \{9+20, 10+15\} = (25)$

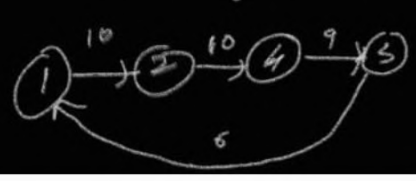
$g(3,\{2,4\}) = \min_{j \in \{2,4\}} \{ C_{32} + g(2,4), C_{34} + g(4,2) \} = \{13+18, 12+13\} = (25)$

$g(4,\{2,3\}) = \min_{j \in \{2,3\}} \{ C_{42} + g(2,3), C_{43} + g(3,2) \} = \{8+15, 9+18\} = (23)$

S=3

$g(1,\{2,3,4\}) = \min_{j \in \{2,3,4\}} \{ C_{12} + g(2,\{3,4\}), C_{13} + g(3,\{2,4\}), C_{14} + g(4,\{2,3\}) \}$

$= \{10+25, 15+25, 20+23\} = (35)$



1 2 3 4 5

— RELIABILITY DESIGN

$\sum C_i = C_1 + C_2 + C_3$

$= 30 + 15 + 20 = \underline{65}$

Remaining money,

$C - \sum C_i = 105 - 65 = (40)$

D_i	C_i	r_i	u_i
D_1	30	0.9	2
D_2	15	0.8	3
D_3	20	0.5	3

$C = 105$

$m_i = \left[\frac{C - \sum C_i}{C_i}\right] + 1$ → no. of copies of each device (1+ extras)

R C

$S^0 = \{(1, 0)\}$

Consider D_1,

$S_1^1 = \{(0.9, 30)\}$

$S_2^1 = \{(0.99, 60)\}$

$1-(1-r_1)^2$

$= 1-(1-0.9)^2$

$= 1-(0.1)^2$

$= 1-0.01$

$= (0.99)$

$S^1 = \{(0.9, 30), (0.99, 60)\}$ — (1)

Consider D_2, $S_1^2 = \{(0.72, 45), (0.792, 75)\}$

Reliability of D_2 is multiplied with (1) eq

cost of D_2 is added to (1)

$S_2^2 = \{(0.864, 60), (\cancel{0.9504, 90}\}$

(0.9×0.96) $(30+30)$ (0.99×0.96) $(60+30)$

$1-(1-r_2)^2$

$1-(1-0.8)^2$

$1-(0.2)^2$

$1-(0.04)$

$= 0.96$ 30

this is not feasible as the cost exceeds that if is not possible to buy device 3

$S_3^2 = \{(0.8928, 75), (\cancel{\quad, 105}) \}$

0.9×0.992 ; $30 + 45$

$1-(1-\gamma_2)^3$

$1-(1-0.8)^3$

$1-(0.2)^3$

$1-(0.008)$

$= 0.992$ 45

$S_.^2 = \{(0.72, 45), \cancel{(0.792, 75)}, (0.864, 60), (0.8928, 75)\}$ — (2)

cost is increasing but reliability is decreasing do, it is removed.

Consider D_3,

$S_1^3 = \{(0.36, 65), (0.432, 80), (0.4464, 95)\}$

0.75×0.5 ; $45+20$ ... like that with (2) and c_3 and γ_3

$S_2^3 = \{(0.54, 85), (0.648, 100), (\cancel{\quad, 115})\}$

0.72×0.75 ; $45+40$ ... do on

$\gamma_3 = 0.5$

$1-(1-\gamma_3)^2$

$1-(1-0.5)^2$

$1-0.25$

0.75 40

$S_3^2 = \{(0.63, 105), (\cancel{\quad, 120}) (\cancel{\quad, 125})\}$

$1-(1-\gamma_3)^3$

$S^3 = \{(0.36, 65) (0.432, 80), (\cancel{0.4464, 95}),$

$1-(1-0.5)^3$

$1-0.125$

$\{(0.54, 85), (0.63, 105), (0.648, 100)\}$ (0.875) 60

Ans: Max reliability → 0.648 cost → 100

$D_3 \rightarrow$ 2 copies as it is present at S_2^3

$D_2 \rightarrow$ 2 copies as it is present at S_2^2

$D_1 \rightarrow$ 1 copy as it is present at S_1^1

Ans →

D_1	D_2	D_3
1	2	2

$\pi r_i = 0.648$

$\Sigma c_i = 100$

- OPTIMAL BINARY SEARCH TREE (OBST)

Mostly leaving X