

Branch and Bound (Introduction)

Branch and Bound



Job sequencing.

$$\text{Jobs} = \{J_1, J_2, J_3, J_4\}$$

$$P = \{10, 5, 8, 3\}$$

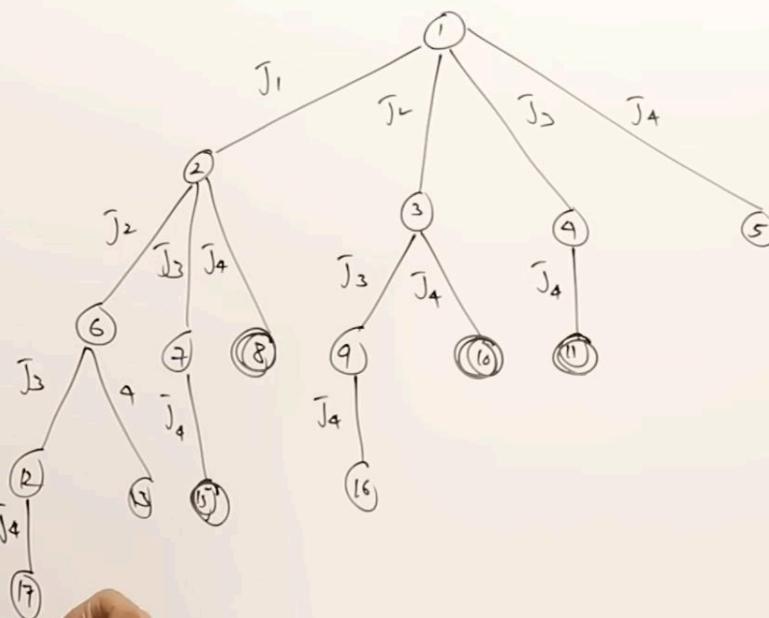
$$d = \{1, 2, 1, 2\}$$

$$S1 = \{J_1, J_4\}$$

$$S2 = \{1, 0, 0, 1\}$$



Branch and Bound



Job sequencing.

$$\text{Jobs} = \{J_1, J_2, J_3, J_4\}$$

$$P = \{10, 5, 8, 3\}$$

$$d = \{1, 2, 1, 2\}$$

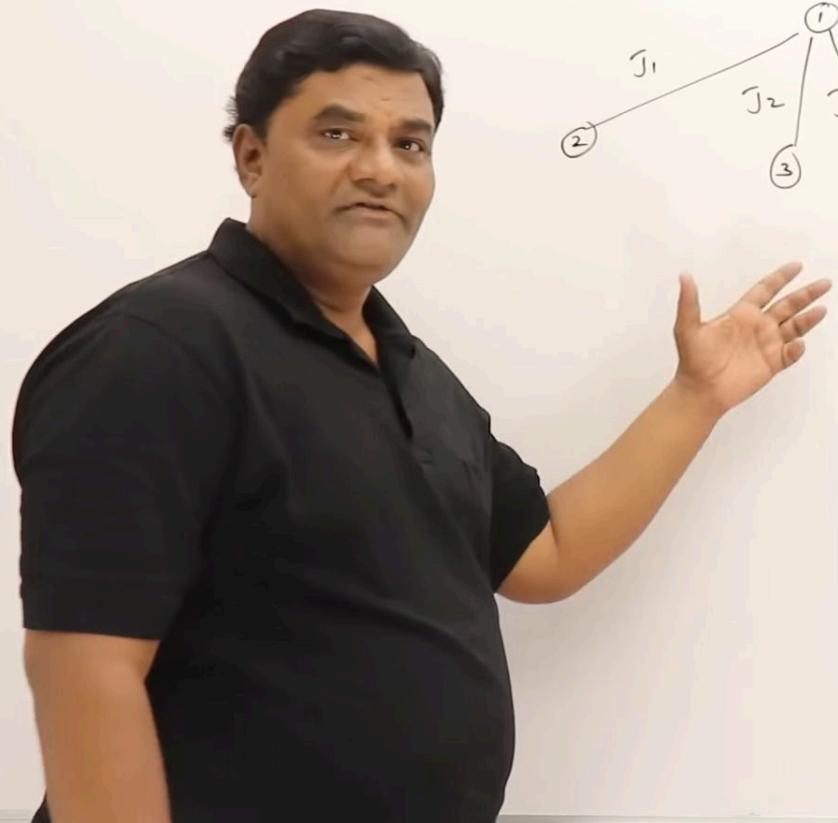
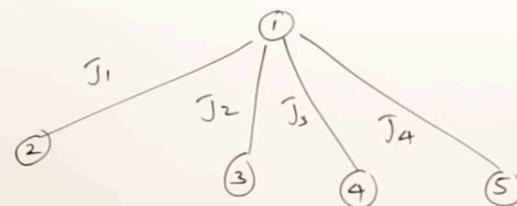
✓

$$S1 = \{J_1, J_4\}$$

$$S2 = \{1, 0, 0, 1\}$$



Branch and Bound



Job sequencing

$$\text{Jobs} = \{J_1, J_2, J_3, J_4\}$$

$$P = \{10, 5, 8, 3\}$$

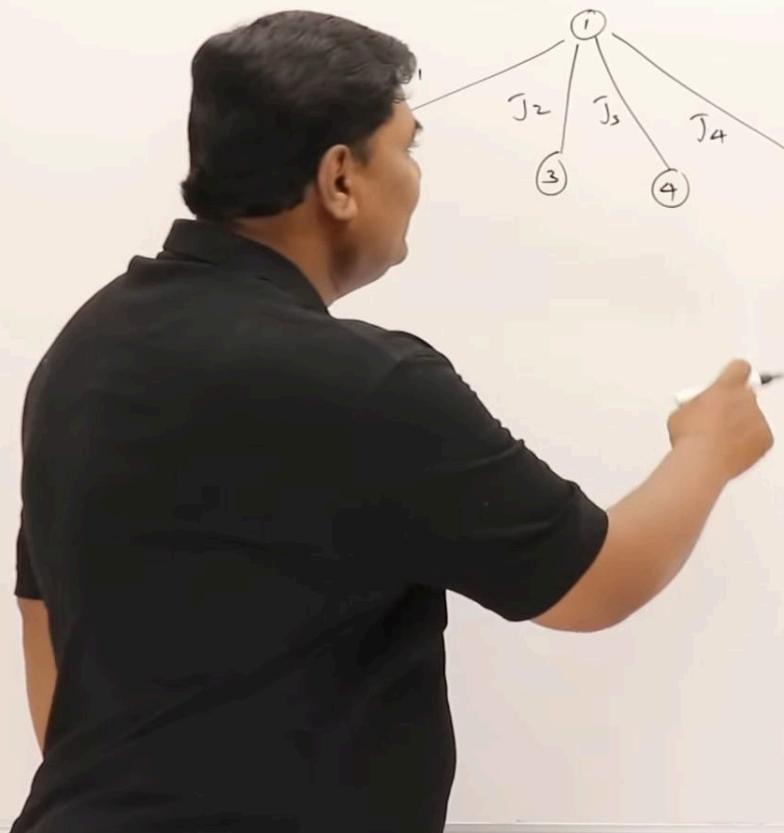
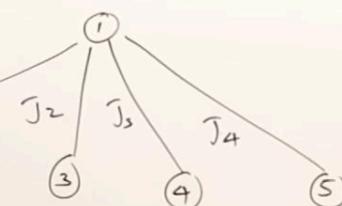
$$d = \{1, 2, 1, 2\}$$

$$S_1 = \{J_1, J_4\}$$

5
4
3
2



Branch and Bound



Job sequencing

$$\text{Jobs} = \{J_1, J_2, J_3, J_4\}$$

$$P = \{10, 5, 8, 3\}$$

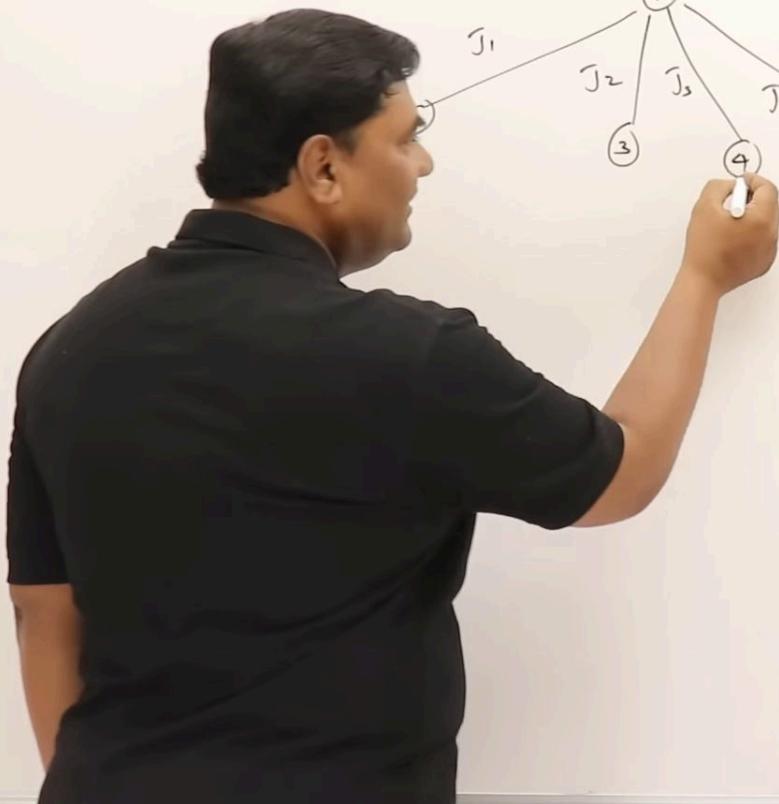
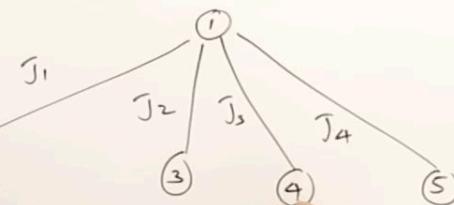
$$d = \{1, 2, 1, 2\}$$

$$S_1 = \{J_1, J_4\}$$

4
3
2



Branch and Bound



Job sequencing

$$\text{Jobs} = \{J_1, J_2, J_3, J_4\}$$

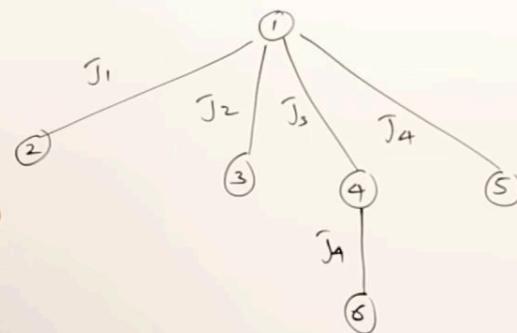
$$P = \{10, 5, 8, 3\}$$

$$d = \{1, 2, 1, 2\}$$

$$S_1 = \{J_1, J_4\}$$



Branch and Bound



Job sequencing

$$\text{Jobs} = \{J_1, J_2, J_3, J_4\}$$

$$P = \{10, 5, 8, 3\}$$

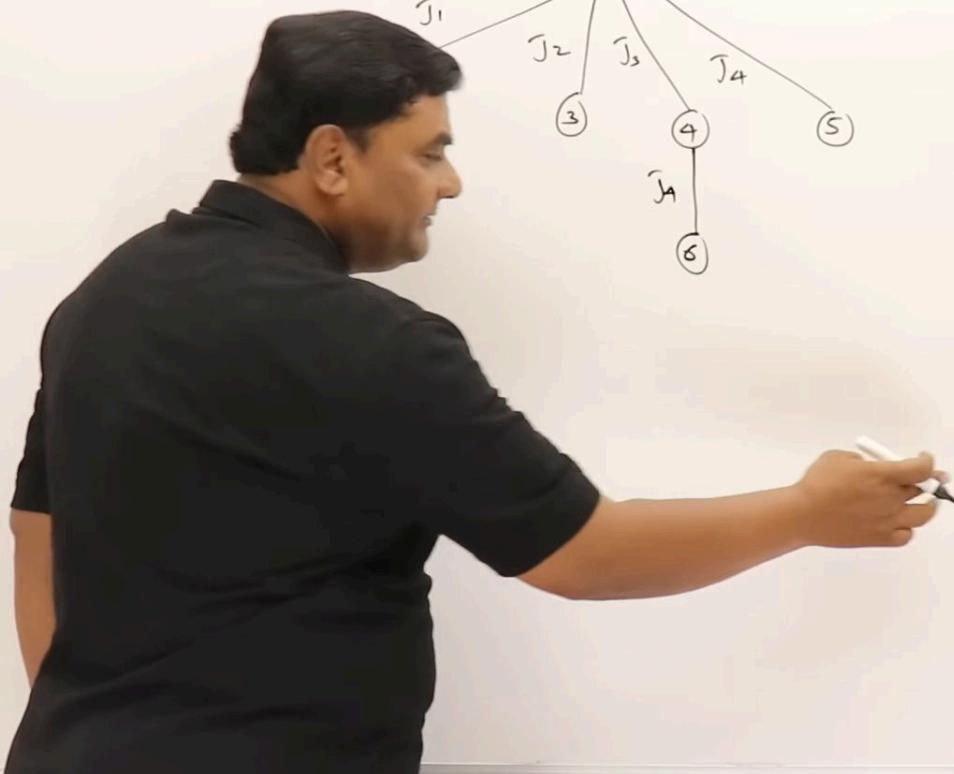
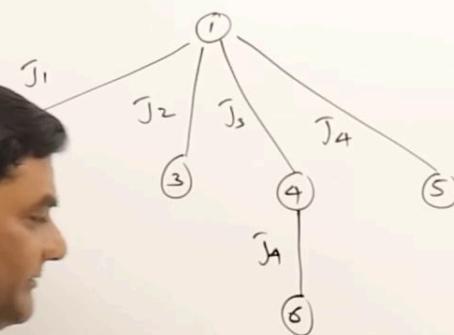
$$d = \{1, 2, 1, 2\}$$

$$S_1 = \{J_1, J_4\}$$

6
3
2



Branch and Bound



Job sequencing

$$\text{Jobs} = \{J_1, J_2, J_3, J_4\}$$

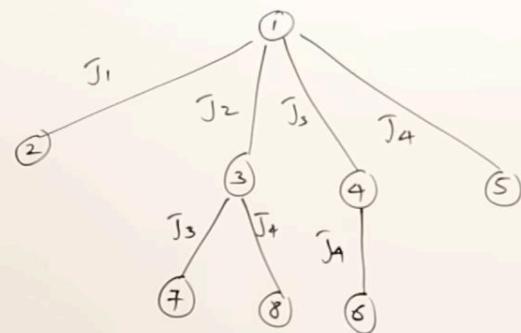
$$P = \{10, 5, 8, 3\}$$

$$d = \{1, 2, 1, 2\}$$

$$S_1 = \{J_1, J_4\}$$



Branch and Bound



Job sequencing

$$\text{Jobs} = \{J_1, J_2, J_3, J_4\}$$

$$P = \{10, 5, 8, 3\}$$

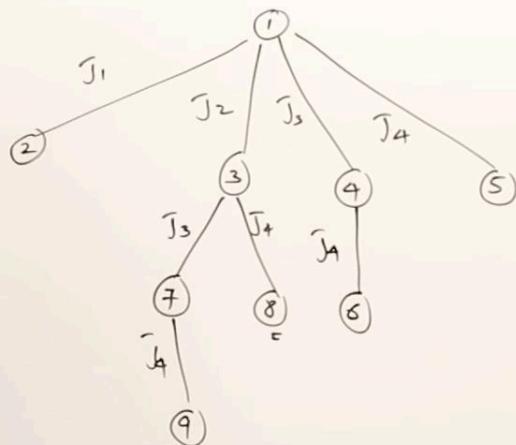
$$d = \{1, 2, 1, 2\}$$

$$S_1 = \{J_1, J_4\}$$

8
7
2



Branch and Bound



Job sequencing

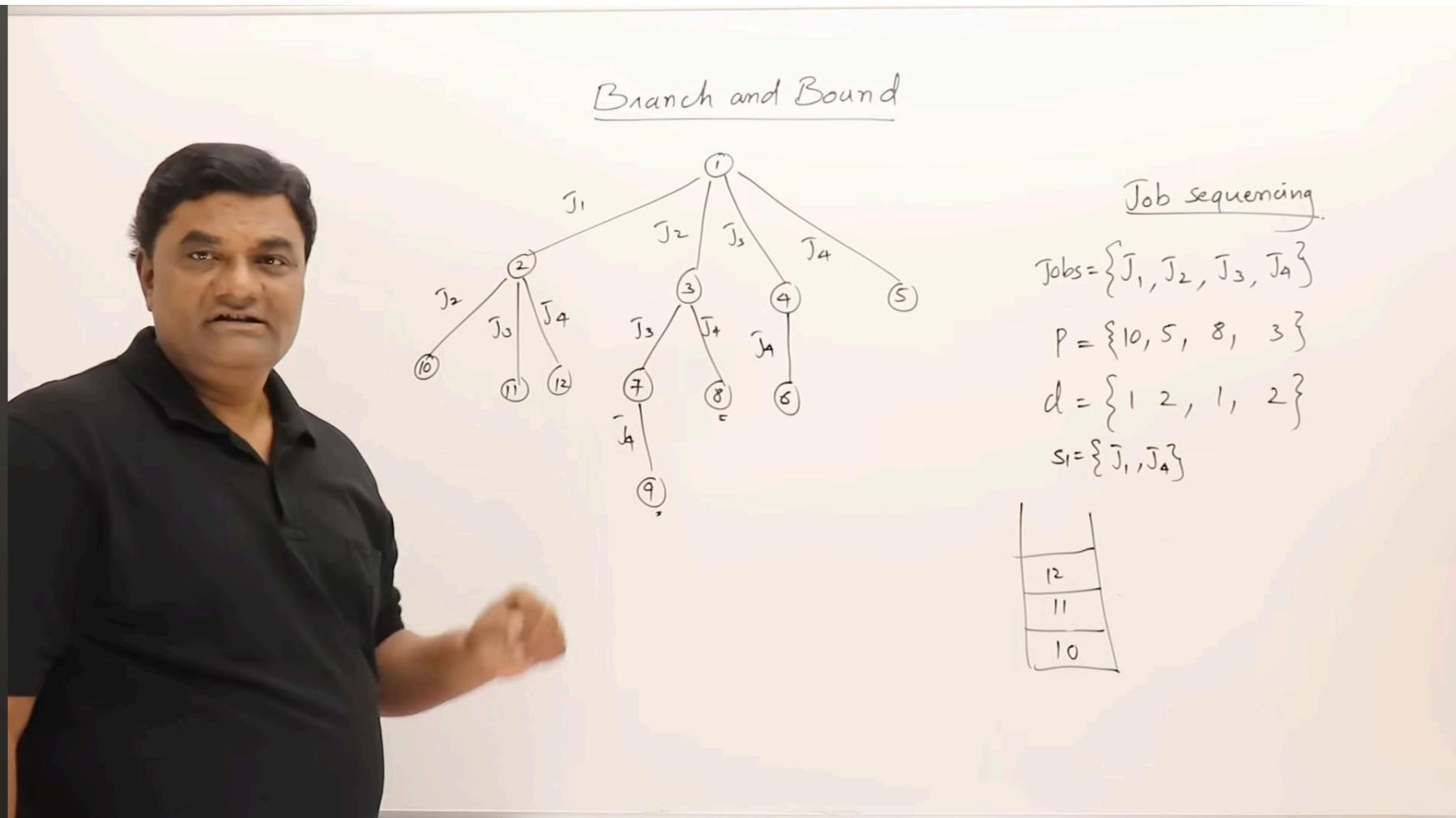
$$\text{Jobs} = \{J_1, J_2, J_3, J_4\}$$

$$P = \{10, 5, 8, 3\}$$

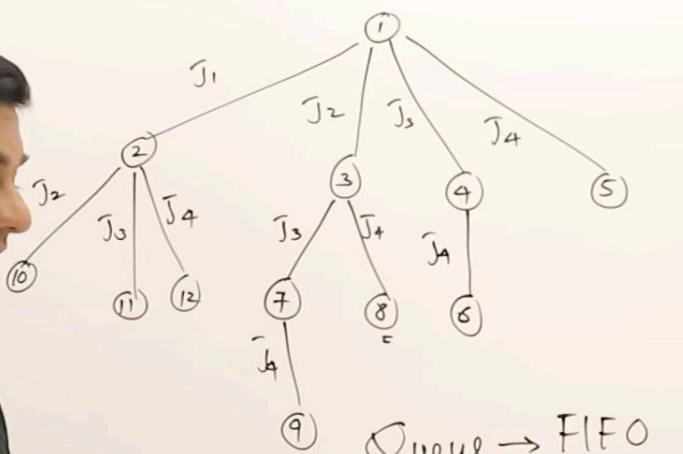
$$d = \{1, 2, 1, 2\}$$

$$S_1 = \{J_1, J_4\}$$





Branch and Bound



Queue \rightarrow FIFO

Stack \rightarrow LIFO

Least cost \rightarrow LC-BB

Job sequencing

$$\text{Jobs} = \{J_1, J_2, J_3, J_4\}$$

$$P = \{10, 5, 8, 3\}$$

$$d = \{1, 2, 1, 2\}$$

$$S_1 = \{J_1, J_4\}$$

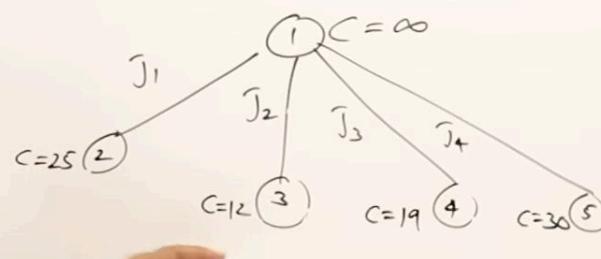


Remember B&B is always BFS, whereas recursion is DFS



Branch and Bound

Least Cost - BB



Job sequencing

$$\text{Jobs} = \{J_1, J_2, J_3, J_4\}$$

$$P = \{10, 5, 8, 3\}$$

$$d = \{1, 2, 1, 2\}$$

$$S_1 = \{J_1, J_4\}$$

1. FIFO-BB

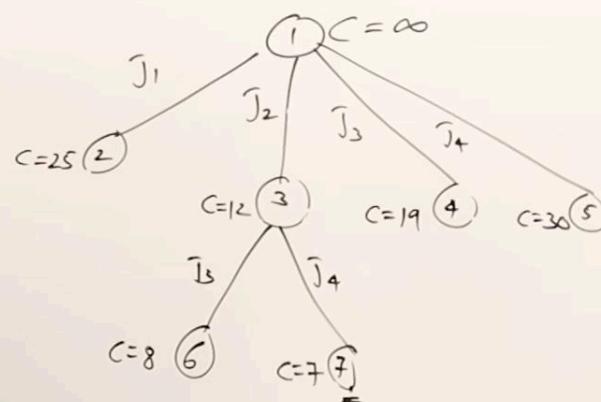
2. LIFO-BB

3. LC-BB



Branch and Bound

Least Cost - BB



Job sequencing

$$\text{Jobs} = \{J_1, J_2, J_3, J_4\}$$

$$P = \{10, 5, 8, 3\}$$

$$d = \{1, 2, 1, 2\}$$

$$S_1 = \{J_1, J_4\}$$

1. FIFO-BB

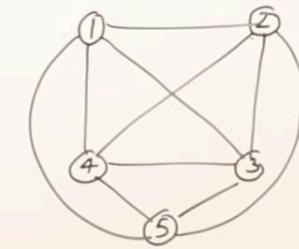
2. LIFO-BB

3. LC-BB



Traveling salesman problem (Using branch and bound)

Traveling Salesperson Branch-n-Bound

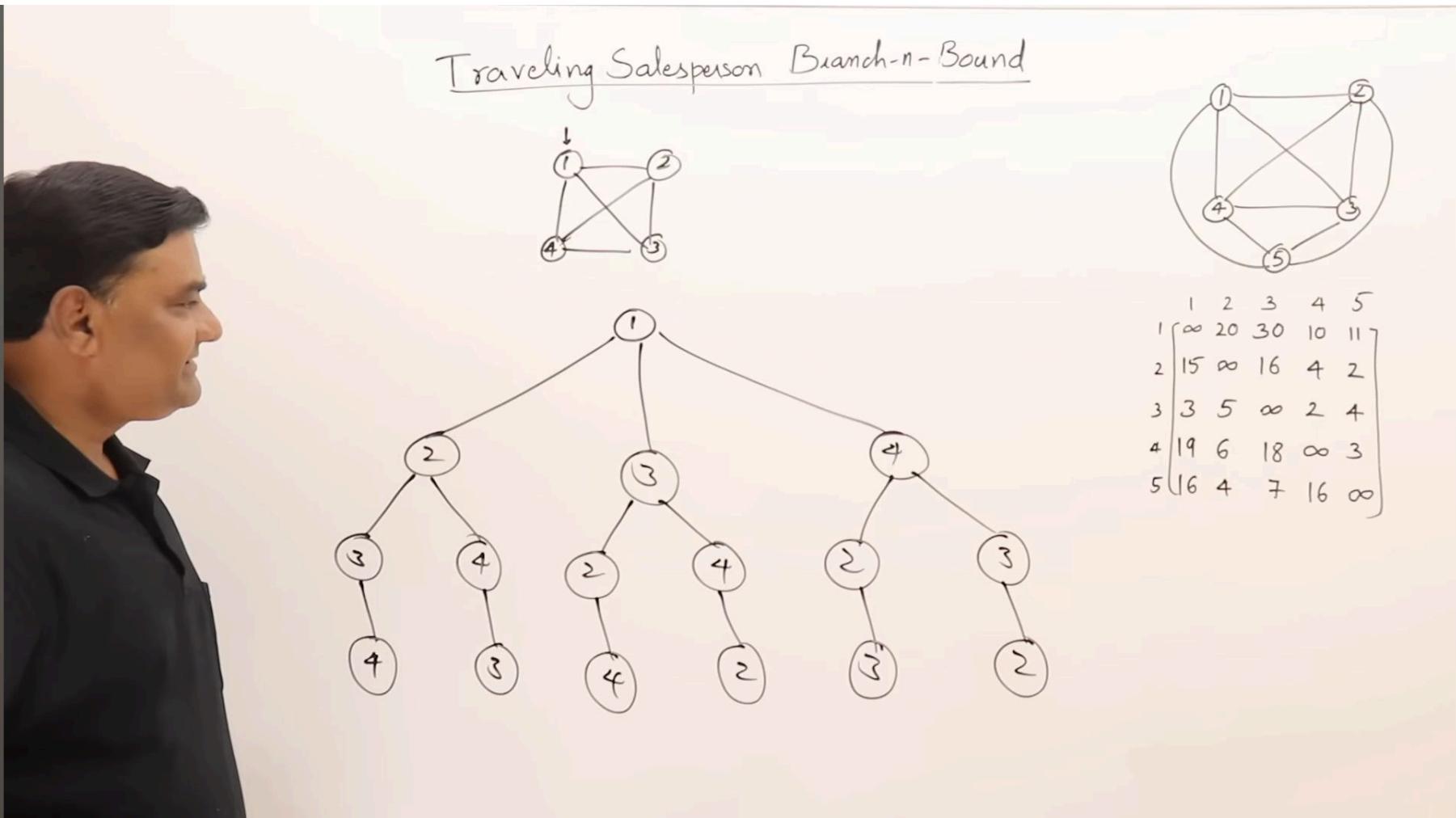


1	2	3	4	5
2	∞	20	30	10
3	15	∞	16	4
4	3	5	∞	2
5	19	6	18	∞

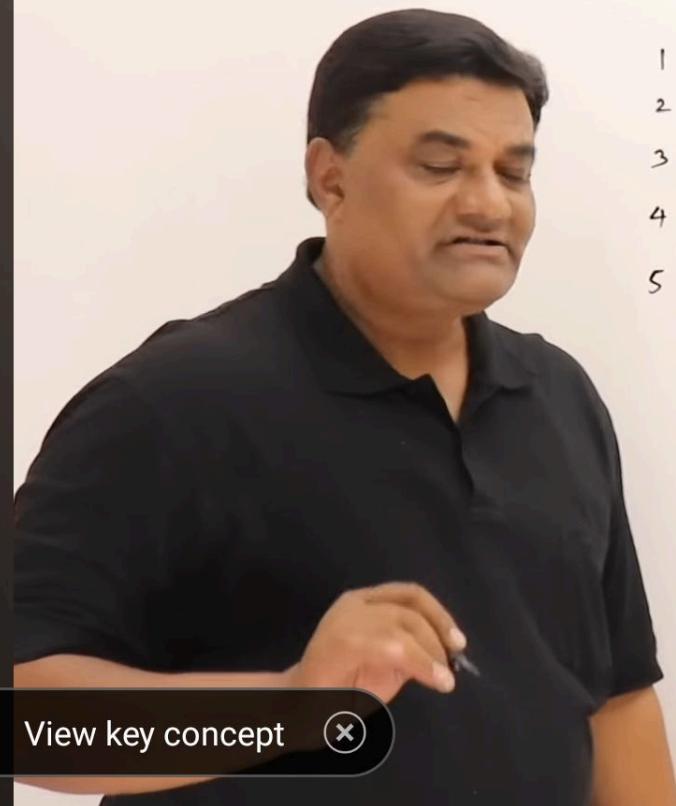


View key concept

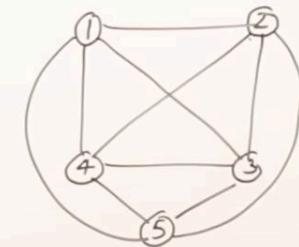




Traveling Salesperson Branch-n-Bound



1	∞	20	30	10	11
2	15	∞	16	4	2
3	3	5	∞	2	4
4	19	6	18	∞	3
5	16	4	7	16	∞



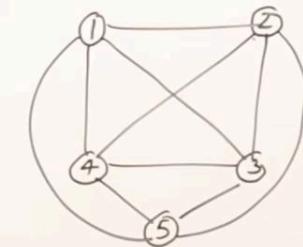
1	2	3	4	5	
1	∞	20	30	10	11
2	15	∞	16	4	2
3	3	5	∞	2	4
4	19	6	18	∞	3
5	16	4	7	16	∞

View key concept



Traveling Salesperson Branch-n-Bound

	1	2	3	4	5	
1	∞	20	30	10	11	10
2	15	∞	16	4	2	2
3	3	5	∞	2	4	2
4	19	6	18	∞	3	3
5	16	4	7	16	∞	4

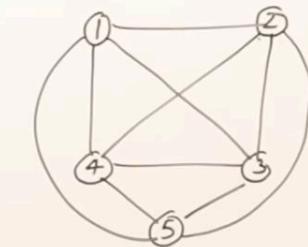


	1	2	3	4	5	
1	∞	20	30	10	11	
2	15	∞	16	4	2	
3	3	5	∞	2	4	
4	19	6	18	∞	3	
5	16	4	7	16	∞	



Traveling Salesperson Branch-n-Bound

1	2	3	4	5	
1	∞	10	20	0	1
2	13	∞	14	2	0
3	1	3	∞	0	2
4	16	3	15	∞	0
5	12	0	3	12	∞
					$\frac{21}{21}$

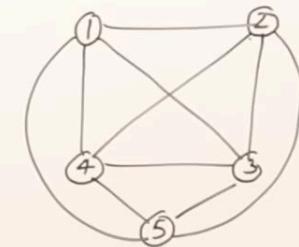


1	2	3	4	5	
1	∞	20	30	10	11
2	15	∞	16	4	2
3	3	5	∞	2	4
4	19	6	18	∞	3
5	16	4	7	16	∞



Traveling Salesperson Branch-n-Bound

	1	2	3	4	5	
1	∞	10	20	0	1	10
2	13	∞	14	2	0	2
3	1	3	∞	0	2	2
4	16	3	15	∞	0	3
5	12	0	3	12	∞	4
	1	0	3	0	0	<u>21</u>

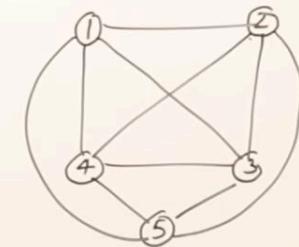


	1	2	3	4	5
1	∞	20	30	10	11
2	15	∞	16	4	2
3	3	5	∞	2	4
4	19	6	18	∞	3
5	16	4	7	16	∞



Traveling Salesperson Branch-n-Bound

$$\begin{array}{cc} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \left[\begin{matrix} \infty & 10 & 17 & 0 & 1 \\ 12 & \infty & 11 & 2 & 0 \\ 0 & 3 & \infty & 0 & 2 \\ 15 & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & 12 & \infty \end{matrix} \right] \begin{matrix} 10 \\ 2 \\ 2 \\ 3 \\ 4 \end{matrix} \\ & 1 \ 0 \ 3 \ 0 \ 0 + \underline{21+9=25} \end{array}$$

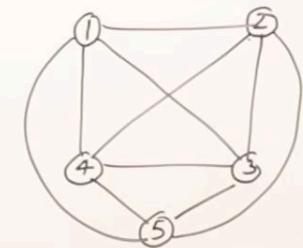


	1	2	3	4	5
1	∞	20	30	10	11
2	15	∞	16	4	2
3	3	5	∞	2	4
4	19	6	18	∞	3
5	16	4	7	16	∞



Traveling Salesperson Branch-n-Bound

1	2	3	4	5
1	∞	10	17	0
2	12	∞	11	2
3	0	3	∞	2
4	15	3	12	∞
5		0	0	12



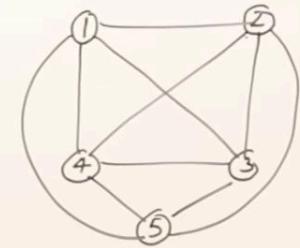
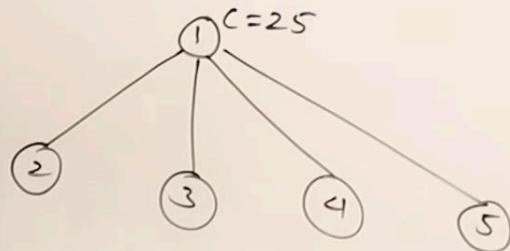
1	2	3	4	5
1	∞	10	17	0
2	12	∞	11	2
3	0	3	∞	2
4	15	3	12	∞
5	11	0	0	12

Reduced cost = 25

Traveling Salesperson Branch-n-Bound

1	2	3	4	5	
1	∞	10	17	0	1
2	12	∞	11	2	0
3	0	3	∞	0	2
4	15	3	12	∞	0
5	11	0	0	12	∞

upper = ∞

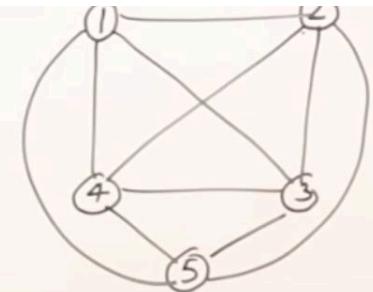
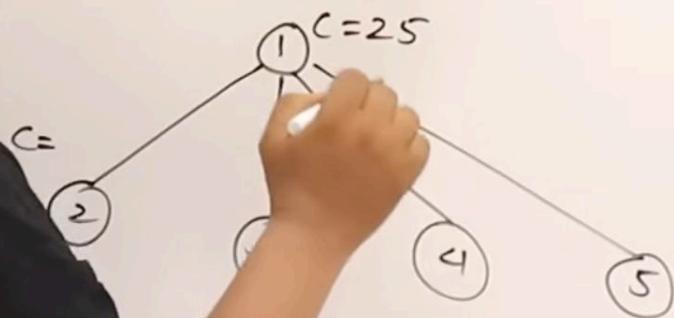


1	2	3	4	5	
1	∞	10	17	0	1
2	12	∞	11	2	0
3	0	3	∞	0	2
4	15	3	12	∞	0
5	11	0	0	12	∞

reduced cost = 25

	1	2	3	4	5
1	0	0	0	0	0
2	0	0	11	2	0
3	0	0	0	0	2
4	15	0	12	0	0
5	11	0	0	12	0

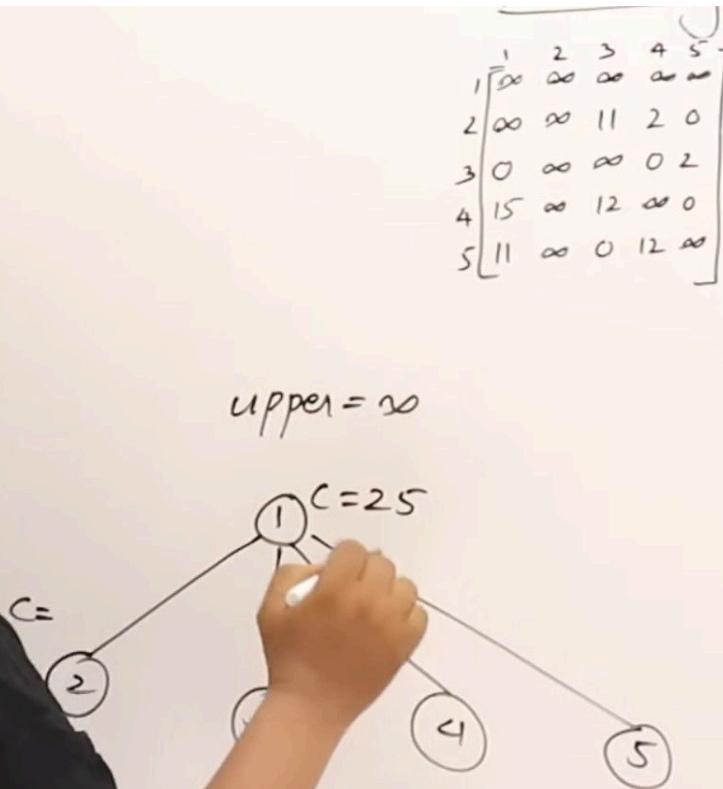
upper = 20



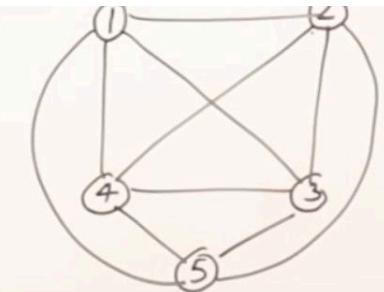
	1	2	3	4	5
1	0	10	17	0	17
2	12	0	11	2	0
3	0	3	0	0	2
4	15	3	12	0	0
5	11	0	0	12	0

reduced cost = 25





After making infinity
check if all rows and
columns have atleast
one 0 or not



	1	2	3	4	5
1	∞	10	17	0	17
2	12	∞	11	2	0
3	0	3	∞	0	2
4	15	3	12	∞	0
5	11	0	0	12	∞

reduced cost = 25



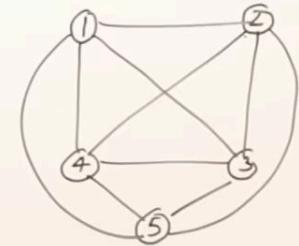
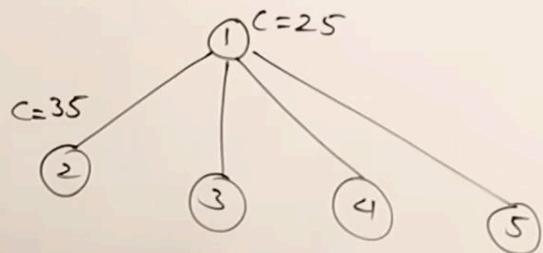
Traveling Salesperson Branch-n-Bound

1	2	3	4	5
1	∞	2	3	4
2	∞	∞	11	2
3	0	∞	∞	0
4	15	∞	12	∞
5	11	∞	0	12

$$C(1, 2) + \gamma + \hat{\gamma}$$

upper1 = ∞

$$10 + 25 + 0$$



1	2	3	4	5
1	∞	10	17	0
2	12	∞	11	2
3	0	3	∞	0
4	15	3	12	∞
5	11	0	0	12

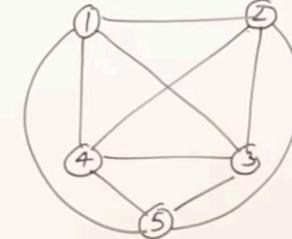
reduced cost = 25



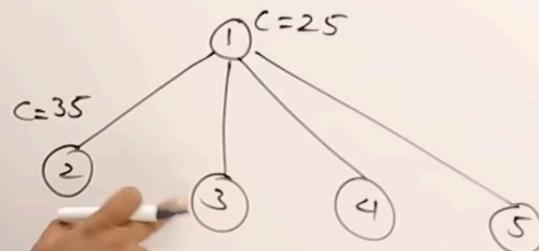
Traveling Salesperson Branch-n-Bound

1	2	3	4	5
1	∞	2	3	4
2	∞	∞	11	2
3	0	∞	∞	2
4	15	∞	12	∞
5	11	∞	0	12

1	2	3	4	5
1	∞	10	17	0
2	12	∞	11	2
3	0	3	∞	2
4	15	3	12	∞
5	11	0	0	12



upper1 = 30



1	2	3	4	5
1	∞	10	17	0
2	12	∞	11	2
3	0	3	∞	2
4	15	3	12	∞
5	11	0	0	12

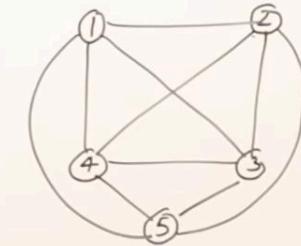
reduced cost = 25



Traveling Salesperson Branch-n-Bound

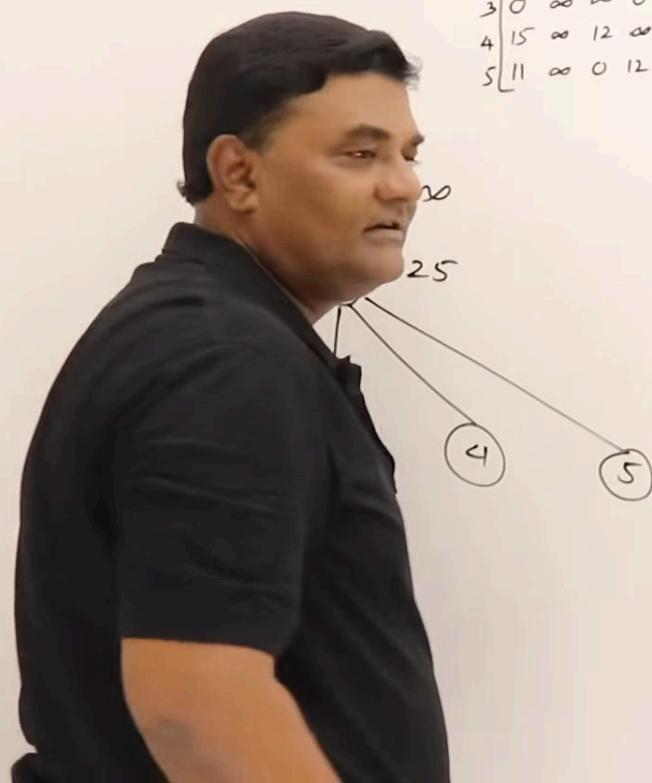
	1	2	3	4	5
1	∞	2	3	4	5
2	∞	∞	11	2	0
3	0	∞	∞	0	2
4	15	∞	12	∞	0
5	11	∞	0	12	∞

	1	2	3	4	5
1	∞	2	3	4	5
2	12	∞	20		
3	20	3	∞	0	2
4	15	3	∞	∞	0
5	11	0	∞	12	0



	1	2	3	4	5
1	∞	10	17	0	17
2	12	∞	11	2	0
3	0	3	∞	0	2
4	15	3	12	∞	0
5	11	0	0	12	∞

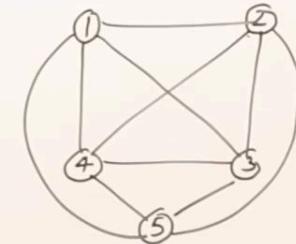
Reduced cost = 25



Traveling Salesperson Branch-n-Bound

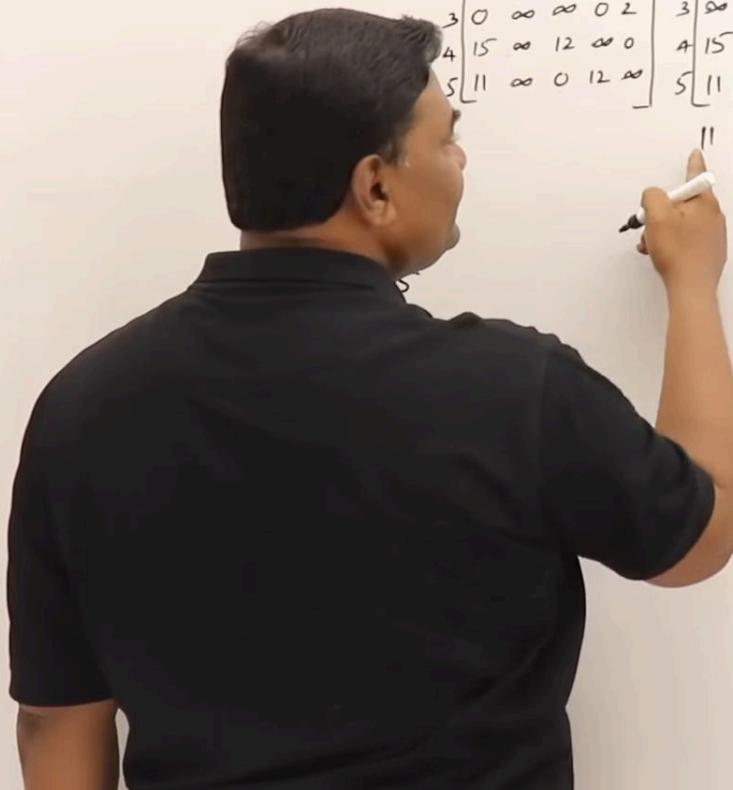
	1	2	3	4	5
1	∞	2	3	4	5
2	∞	∞	11	2	0
3	0	∞	∞	0	2
4	15	∞	12	∞	0
5	11	∞	0	12	∞

11 0 0 0 0



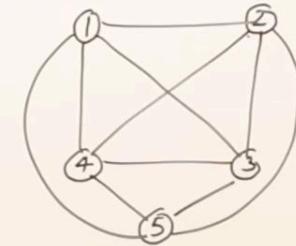
	1	2	3	4	5
1	∞	10	17	0	1
2	12	∞	11	2	0
3	0	3	∞	0	2
4	15	3	12	∞	0
5	11	0	0	12	∞

Reduced cost = 25



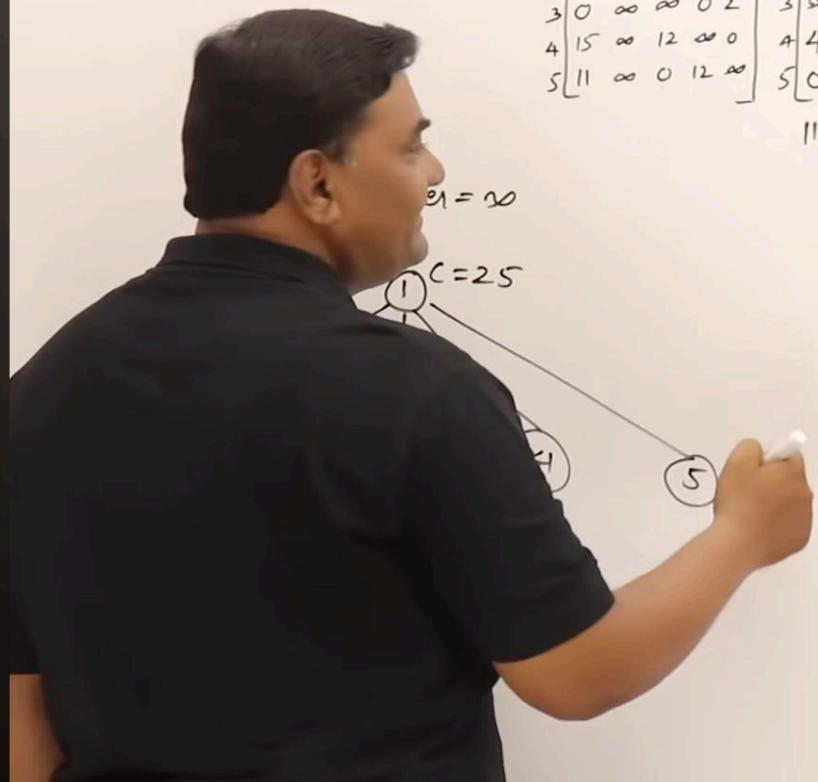
Traveling Salesperson Branch-n-Bound

	1	2	3	4	5
1	∞	2	3	4	5
2	∞	∞	11	2	0
3	0	∞	∞	0	2
4	15	∞	12	∞	0
5	11	∞	0	12	∞
	11	0	0	0	0



	1	2	3	4	5
1	∞	10	17	0	1
2	12	∞	11	2	0
3	0	3	∞	0	2
4	15	3	12	∞	0
5	11	0	0	12	∞

Reduced cost = 25



Traveling Salesperson Branch-n-Bound

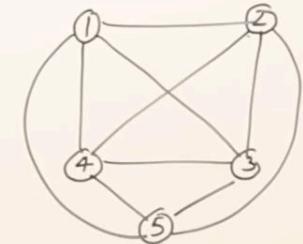
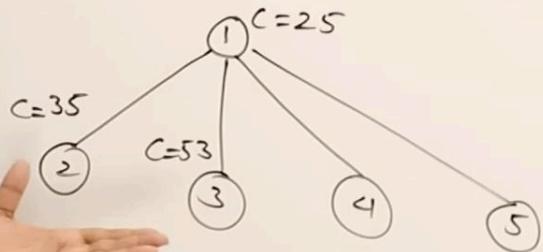
	1	2	3	4	5
1	∞	2	3	4	5
2	∞	∞	11	2	0
3	0	∞	∞	0	2
4	15	∞	12	∞	0
5	11	∞	0	12	∞

11 0 0 0 0

upper1 = ∞

$$C(1,3) + \hat{r} + \hat{r}$$

$$17 + 25 + 11$$



	1	2	3	4	5
1	∞	10	17	0	1
2	12	∞	11	2	0
3	0	3	∞	0	2
4	15	3	12	∞	0
5	11	0	0	12	∞

reduced cost = 25



Traveling Salesperson Branch-n-Bound

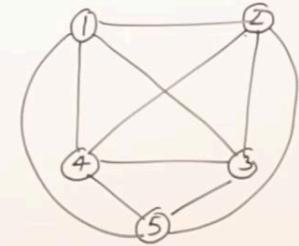
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	2	3	4	5	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	0	0	2	0	2	12	0	11	90
3	0	0	0	0	2	0	3	0	0	0	2	0	3	0	0
4	15	0	12	0	0	4	3	0	0	0	0	40	3	12	00
5	11	0	0	12	00	5	0	0	0	12	0	5	11	0	0

(2)

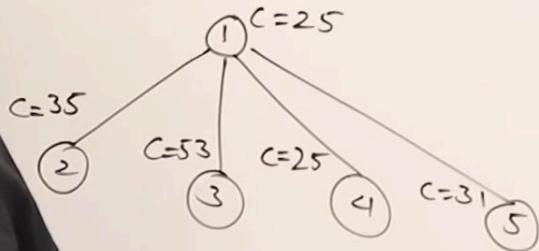
(3)

(4)

(5)



upper1 = 30



	1	2	3	4	5
1	0	10	17	0	17
2	12	0	11	2	0
3	0	3	0	0	2
4	15	3	12	0	0
5	11	0	0	12	00

Reduced cost = 25

Traveling Salesperson Branch-n-Bound

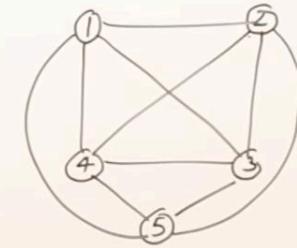
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	∞	2	3	4	5	1	∞	2	3	4	1	∞	2	3	4
2	0	∞	11	2	0	2	1	∞	2	0	2	10	∞	9	0
3	0	∞	∞	0	2	3	0	3	∞	0	2	3	0	3	∞
4	15	∞	12	∞	0	4	4	3	∞	0	0	4	12	0	9
5	11	∞	0	12	∞	5	0	0	∞	12	0	5	∞	0	0

(2)

(3)

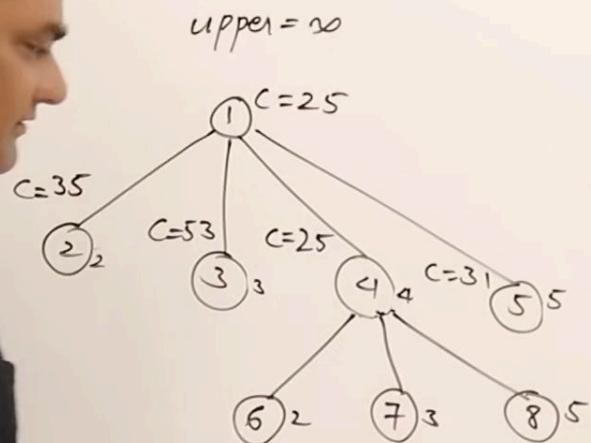
(4)

(5)



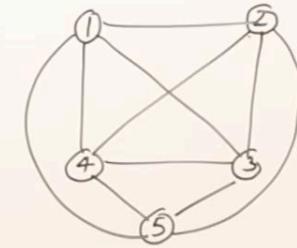
	1	2	3	4	5
1	∞	10	17	0	17
2	12	∞	11	2	0
3	0	3	∞	0	2
4	15	3	12	∞	0
5	11	0	0	12	∞

reduced cost = 25



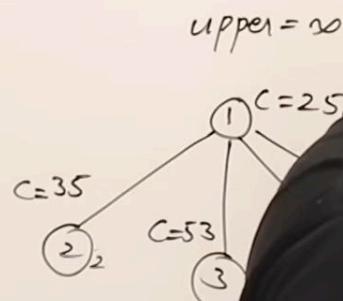
Traveling Salesperson Branch-n-Bound

	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	∞	2	3	4	5	1	∞	2	3	4	1	∞	2	3	4
2	0	∞	11	2	0	2	1	∞	2	0	2	10	∞	9	0
3	0	∞	∞	0	2	3	0	3	∞	0	2	0	3	∞	0
4	4	3	∞	0	0	4	0	3	12	∞	0	12	0	9	∞
5	0	0	∞	12	0	5	11	0	0	∞	0	0	0	12	0



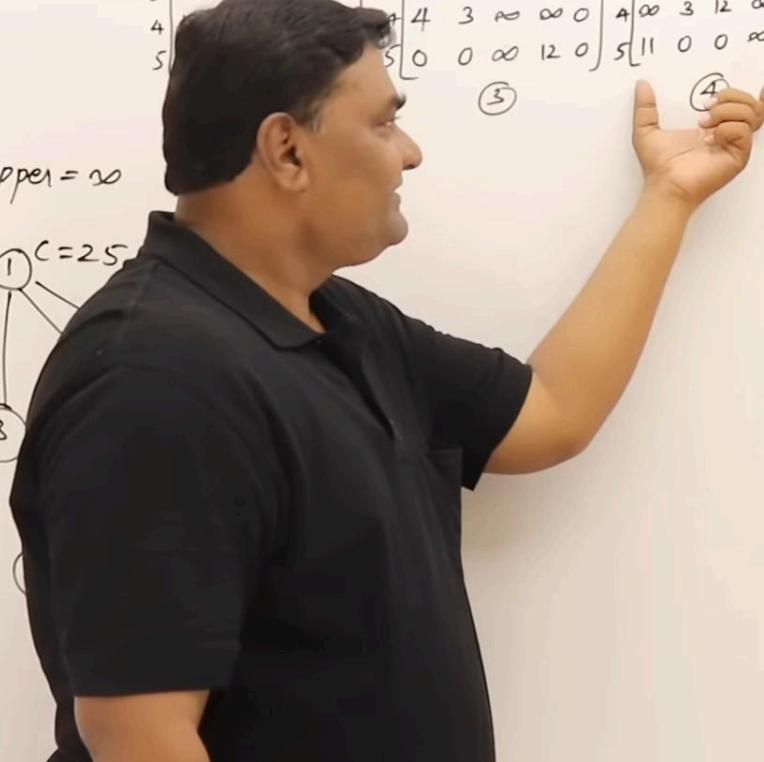
	1	2	3	4	5
1	∞	10	17	0	1
2	12	∞	11	2	0
3	0	3	∞	0	2
4	15	3	12	∞	0
5	11	0	0	12	∞

reduced cost = 25



upper1 = ∞

(3) (4) (5)



Traveling Salesperson Branch-n-Bound

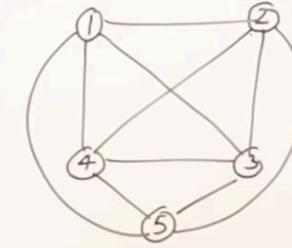
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	2	3	4	5	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	0	0	2	0	12	0	11	0	0
3	0	0	0	0	2	0	3	0	0	2	3	0	3	0	0
4	15	0	12	0	0	4	3	0	0	0	4	0	3	12	0
5	11	0	0	12	0	5	0	0	0	12	5	11	0	0	0

(2)

(3)

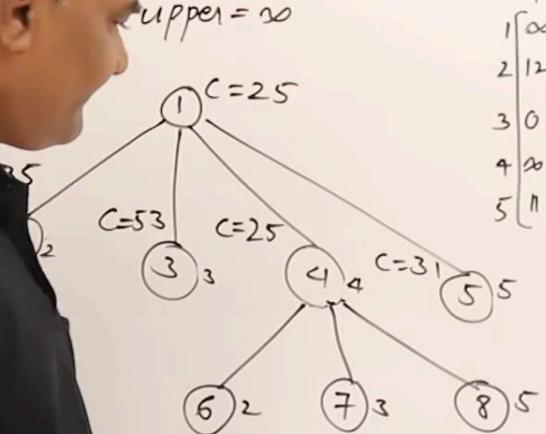
(4)

(5)



	1	2	3	4	5
1	0	10	17	0	17
2	12	0	11	2	0
3	0	3	0	0	2
4	15	3	12	0	0
5	11	0	0	12	0

reduced cost = 25



Traveling Salesperson Branch-n-Bound

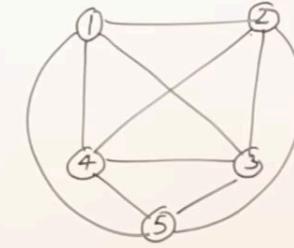
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	2	3	4	5	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	0	0	2	0	2	12	0	11	0
3	0	0	0	0	2	0	3	0	0	2	3	0	3	0	0
4	15	0	0	12	0	4	3	0	0	0	4	0	3	12	0
5	11	0	0	0	12	0	0	0	0	0	5	11	0	0	12

(2)

(3)

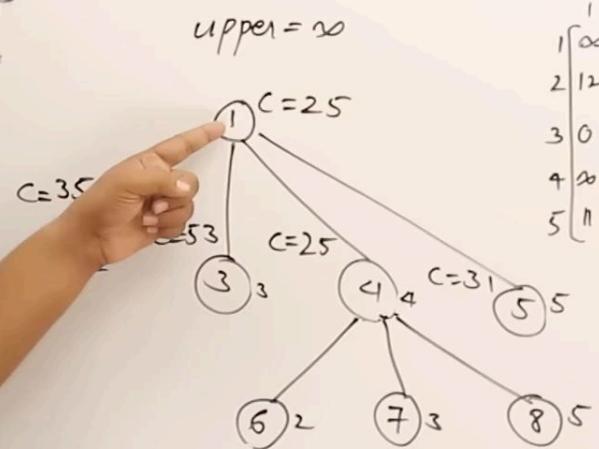
(4)

(5)



	1	2	3	4	5
1	0	10	17	0	17
2	12	0	11	2	0
3	0	3	0	0	2
4	15	3	12	0	0
5	11	0	0	12	0

Reduced cost = 25



Traveling Salesperson Branch-n-Bound

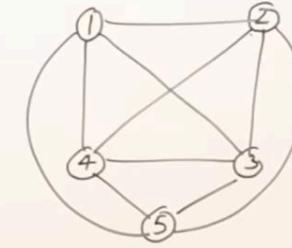
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	2	3	4	5	0	0	0	0	0	0	0	0	0	0
2	0	0	0	11	2	0	1	0	0	2	0	12	0	11	0
3	0	0	0	0	2	0	0	3	0	0	2	0	0	3	0
4	15	0	12	0	0	4	3	0	0	0	4	0	3	12	0
5	11	0	0	12	0	5	0	0	0	12	5	11	0	0	0

(2)

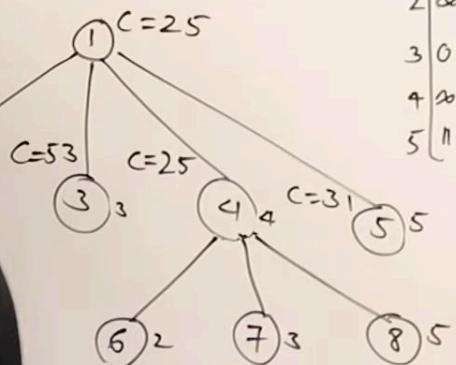
(3)

(4)

(5)



upper1 = 20



	1	2	3	4	5
1	0	0	0	0	0
2	0	0	11	0	0
3	0	0	0	0	2
4	0	0	0	0	0
5	11	0	0	0	0

	1	2	3	4	5
1	0	10	17	0	17
2	12	0	11	2	0
3	0	3	0	0	2
4	15	3	12	0	0
5	11	0	0	12	0

reduced cost = 25



Traveling Salesperson Branch-n-Bound

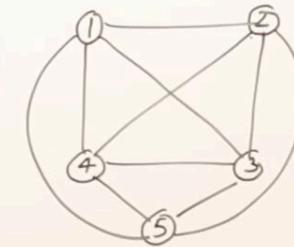
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	∞	∞	∞	∞	0	∞	∞	∞	∞	0	∞	∞	∞	∞
2	∞	0	11	2	0	1	0	∞	2	0	12	0	11	0	0
3	0	∞	0	0	2	∞	3	0	0	2	0	3	0	0	0
4	15	∞	12	0	0	4	3	∞	0	0	4	12	0	9	0
5	11	∞	0	12	∞	5	0	0	∞	12	5	∞	0	0	12

②

③

④

⑤



	1	2	3	4	5
1	0	∞	∞	∞	∞
2	∞	0	11	∞	0
3	0	∞	0	0	2
4	∞	0	0	0	0
5	11	0	0	0	0

C=31

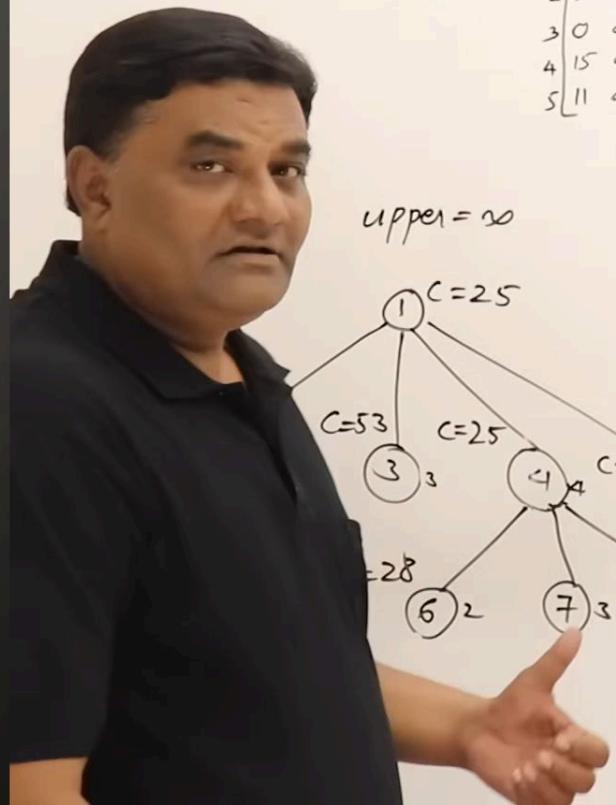
0

0

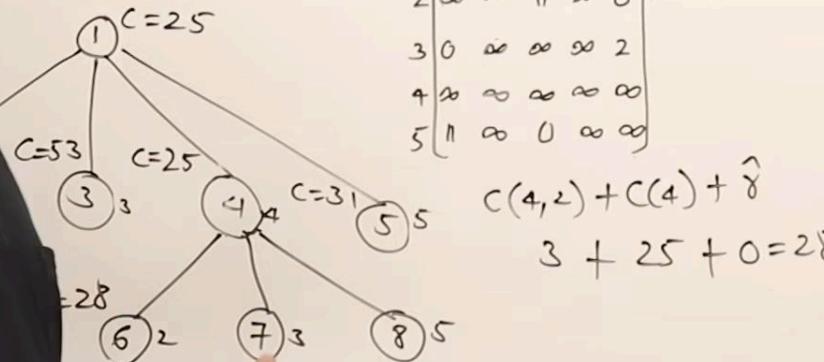
	1	2	3	4	5
1	∞	10	17	0	17
2	12	0	11	2	0
3	0	3	∞	0	2
4	15	3	12	0	0
5	11	0	0	12	0

Reduced cost = 25





upper1 = ∞



Traveling Salesperson Branch-n-Bound

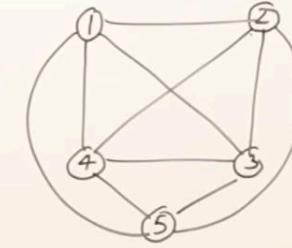
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	∞	2	3	4	5	∞	0	0	0	0	∞	0	0	0	0
2	0	∞	11	2	0	2	1	∞	0	0	12	00	11	00	0
3	0	∞	0	2	0	3	0	∞	0	2	3	0	3	∞	0
4	15	∞	12	∞	0	4	3	∞	0	0	4	00	3	12	∞
5	11	∞	0	12	∞	5	0	∞	12	0	5	11	0	0	∞

(2)

(3)

(4)

(5)



	1	2	3	4	5
1	∞	10	17	0	17
2	12	∞	11	2	0
3	0	3	∞	0	2
4	15	3	12	∞	0
5	11	0	0	12	∞

reduced cost = 25

$$C(4,2) + C(4) + \hat{8}$$

$$3 + 25 + 0 = 28$$



Traveling Salesperson Branch-n-Bound

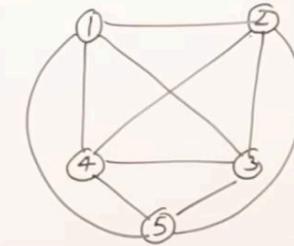
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	∞	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	11	2	1	0	0	2	0	12	0	11	0	0
3	0	0	0	0	2	3	0	3	0	2	3	0	3	0	0
4	15	∞	12	0	0	4	3	0	0	0	4	0	3	12	0
5	11	∞	0	12	∞	5	0	0	0	12	5	11	0	0	12

(2)

(3)

(4)

(5)



upper1 = 20

C=25

C=25

C=31

5

4

2

7

3

8

5

	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	∞	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	11	0	0	2	1	0	0	0	2	1	0	0	0
3	0	0	0	0	2	3	0	1	0	0	3	0	3	0	0
4	0	0	0	0	0	4	0	0	0	0	4	0	0	0	0
5	11	0	0	0	0	5	0	0	0	0	5	0	0	0	0

(6)

(7)

(8)

reduced cost = 25

	1	2	3	4	5
1	0	∞	0	0	0
2	0	0	11	0	0
3	0	0	0	0	2
4	0	0	0	0	0
5	11	0	0	0	0

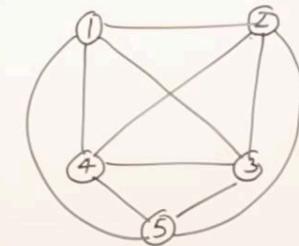


Traveling Salesperson Branch-n-Bound

	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	2	3	4	5	0	0	0	0	0	0	0	0	0	0
2	0	0	0	11	20	2	1	00	00	20	2	12	00	11	00
3	0	0	0	0	2	3	00	3	00	02	3	03	00	02	02
4	15	00	12	00	0	4	4	2	00	00	4	00	3	12	00
5	11	00	0	12	00	5	11	00	00	00	5	00	00	12	00

(2)

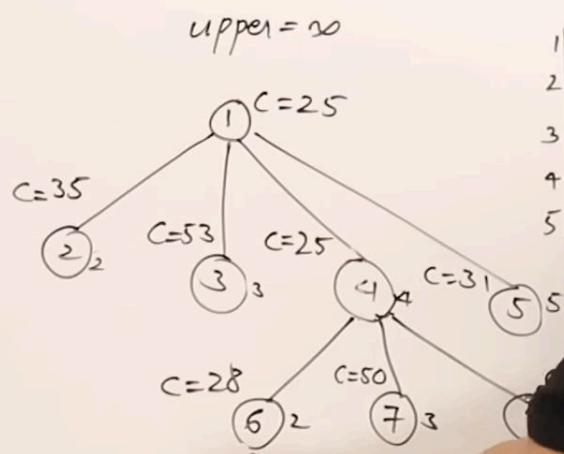
(4) (5)



	2	3	4	5	1	2	3	4	5	1	2	3	4	5
2	0	00	00	00	00	00	00	00	00	00	00	00	00	00
3	00	0	00	00	00	00	00	00	00	00	00	00	00	00
4	00	00	0	00	00	00	00	00	00	00	00	00	00	00
5	00	00	00	0	00	00	00	00	00	00	00	00	00	00

(8)

reduced cost = 25



Traveling Salesperson Branch-n-Bound

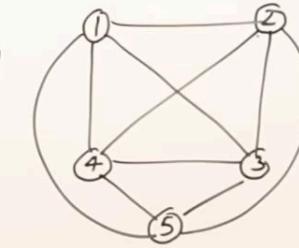
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
2	2	1	0	0	0	2	1	0	0	0	2	10	0	9	0
3	3	0	1	0	0	3	0	3	0	0	3	0	3	0	0
4	4	15	0	12	0	4	4	3	0	0	4	12	0	9	0
5	5	11	0	0	12	5	0	0	0	12	5	0	0	0	12

(2)

(3)

(4)

(5)



	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
2	2	1	0	0	0	2	1	0	0	0	2	12	0	11	2
3	3	0	1	0	0	3	0	3	0	0	3	0	3	0	0
4	4	15	3	12	0	4	4	3	0	0	4	15	3	12	0
5	5	11	0	0	12	5	0	0	0	12	5	0	0	0	12

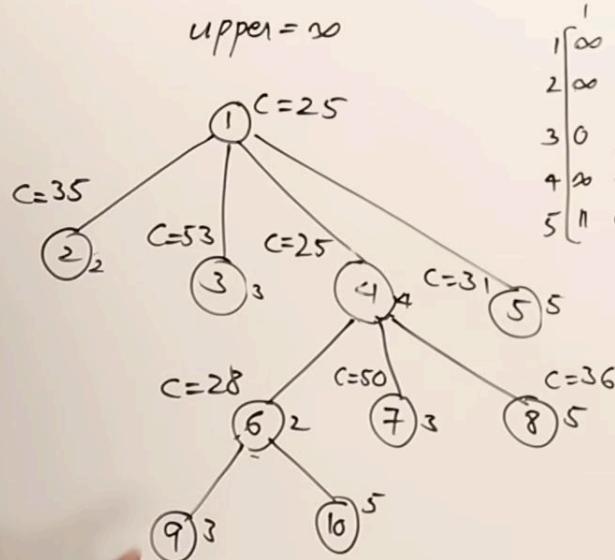
(6)

(7)

(8)

reduced cost = 25

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	0	0	0
3	3	0	1	0	0
4	4	15	3	12	0
5	5	11	0	0	12



Traveling Salesperson Branch-n-Bound

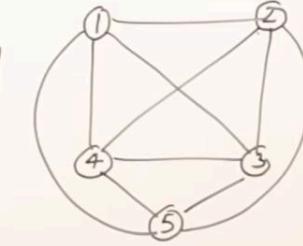
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	2	3	4	5	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	2	0	0	3	0	0	0	0	0	0	0
4	15	0	0	12	0	0	4	3	0	0	0	0	0	0	0
5	11	0	0	0	12	0	0	0	0	0	0	0	0	0	0

(2)

(3)

(4)

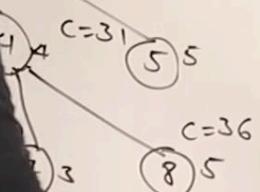
(5)



	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	2	0	0	1	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(6) (7) (8)

Reduced cost = 25



	1	2	3	4	5
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	2
4	0	0	0	0	0
5	11	0	0	0	0



Traveling Salesperson Branch-n-Bound

	1	2	3	4	5
1	0	2	3	4	5
2	2	0	0	11	2
3	0	0	0	0	2
4	15	0	0	12	0
5	11	0	0	0	12

(2)

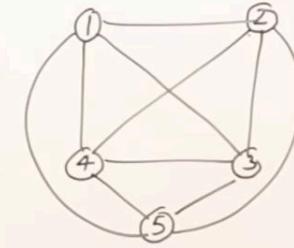
(3)

(4)

(5)

	1	2	3	4	5
1	0	0	0	0	0
2	0	0	0	11	0
3	0	0	0	0	2
4	0	0	0	0	0
5	11	0	0	0	0

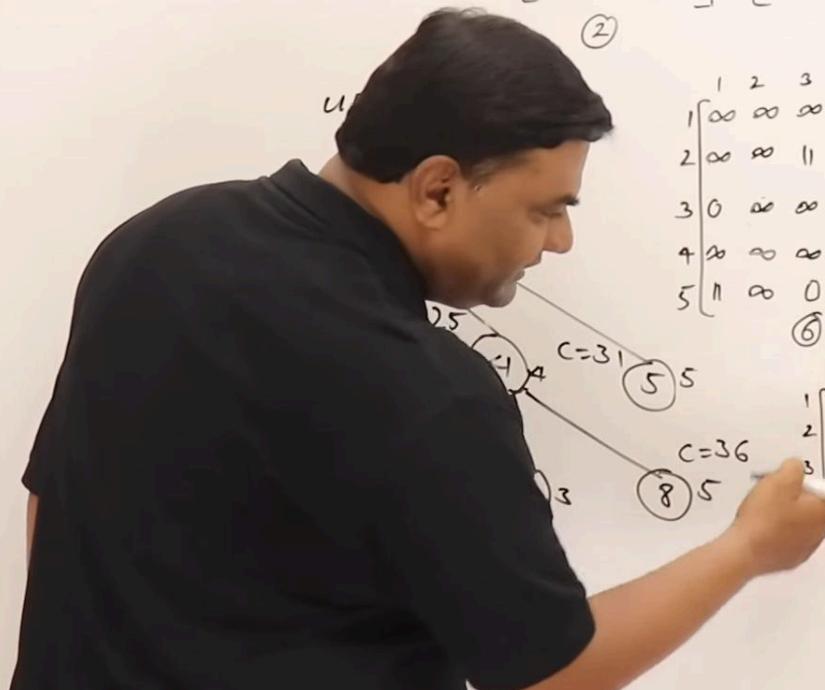
(6) (7) (8)



	1	2	3	4	5
1	0	10	17	0	17
2	12	0	11	2	0
3	0	3	0	0	2
4	15	3	12	0	0
5	11	0	0	12	0

Reduced cost = 25

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



Traveling Salesperson Branch-n-Bound

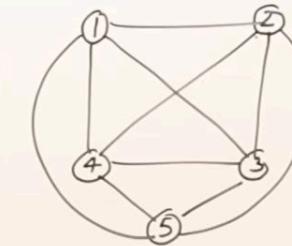
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	2	3	4	5	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	0	0	2	0	2	12	0	11	0
3	0	0	0	0	2	0	3	0	0	2	3	0	3	0	0
4	15	0	12	0	0	4	3	0	0	0	4	0	3	12	0
5	11	0	0	12	0	5	0	0	0	12	5	11	0	0	12

(2)

(3)

(4)

(5)



	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	11	0	0	1	0	0	0	0	2	1	0	0	0
3	0	0	0	0	2	0	1	0	0	0	3	0	3	0	0
4	0	0	0	0	0	4	0	0	0	0	4	0	0	0	0
5	11	0	0	0	0	5	0	0	0	0	5	0	0	0	0

(6)

(7)

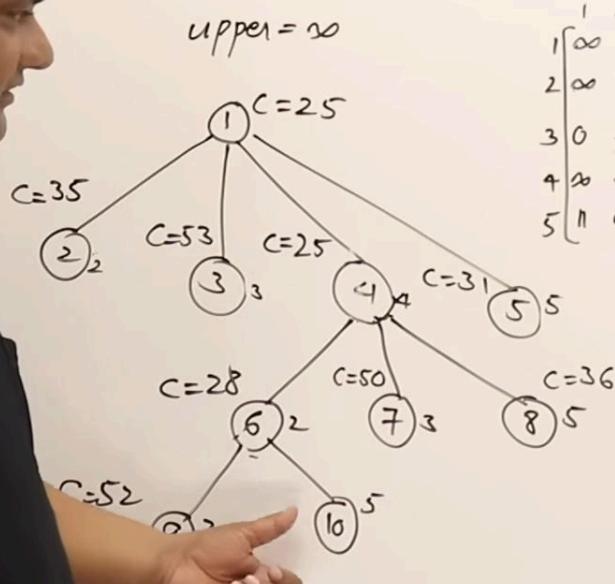
(8)

reduced cost = 25

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

$$C(2,3) + C(6) + 13$$

$$11 + 28 + 13 = 52$$



Traveling Salesperson Branch-n-Bound

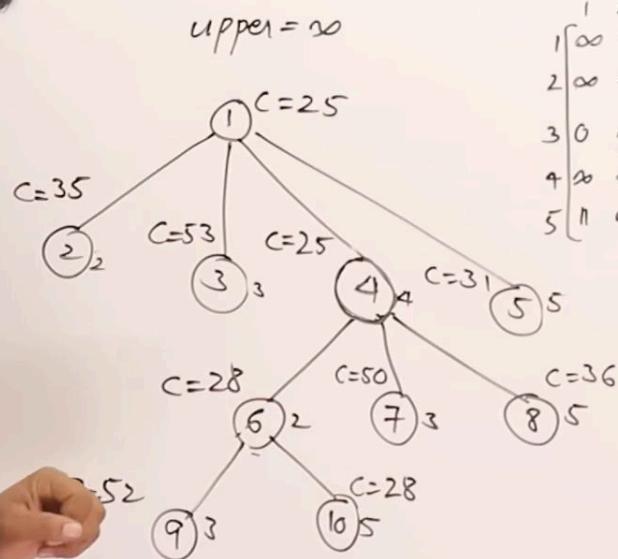
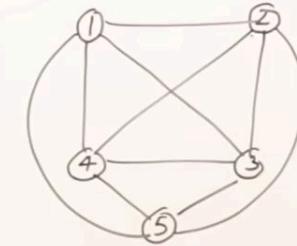
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	∞	2	3	4	5	1	∞	0	0	0	1	∞	0	0	0
2	0	∞	11	2	0	2	1	∞	0	2	2	12	0	11	0
3	0	0	∞	0	2	3	0	3	∞	0	3	0	3	0	0
4	15	∞	12	0	0	4	0	3	12	0	4	12	0	9	0
5	11	∞	0	12	∞	5	0	0	∞	12	5	∞	0	0	12

②

③

④

⑤



	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	∞	0	0	0	0	1	∞	0	0	0	1	∞	0	0	0
2	0	∞	11	0	0	2	1	∞	0	0	2	1	∞	0	0
3	0	0	∞	0	2	3	0	1	∞	0	3	0	3	0	0
4	0	0	0	∞	0	4	0	0	0	0	4	0	0	0	0
5	11	0	0	0	0	5	0	0	0	0	5	0	0	0	0

⑥

⑦

⑧

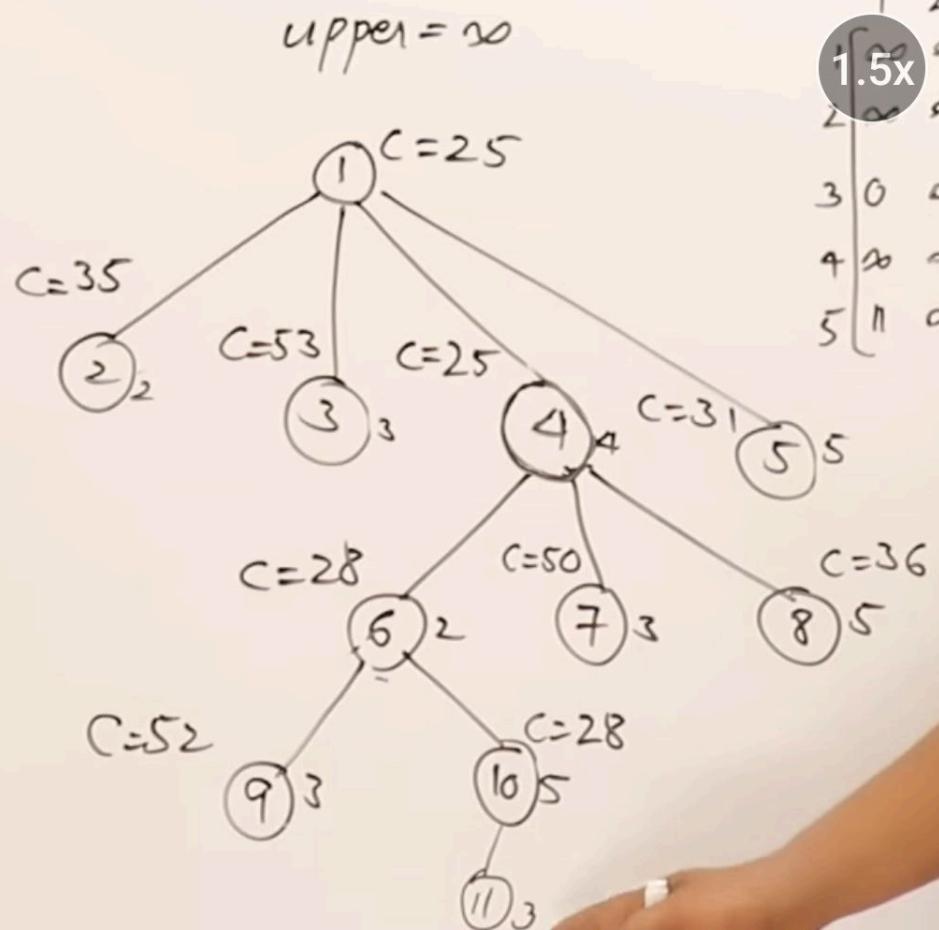
Reduced cost = 25

	1	2	3	4	5	1	2	3	4	5
1	∞	0	0	0	0	1	∞	0	0	0
2	0	∞	0	0	0	2	0	0	0	0
3	0	0	∞	0	0	3	0	20	0	0
4	0	0	0	∞	0	4	0	0	0	0
5	0	0	0	0	0	5	0	0	0	0

⑨

⑩





1.5x

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & \infty & \infty & \infty & \infty \\ 3 & \infty & 11 & \infty & 0 \\ 4 & \infty & \infty & \infty & \infty \\ 5 & 11 & \infty & 0 & \infty \end{bmatrix}$$

⑥

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

⑨



Traveling Salesperson Branch-n-Bound

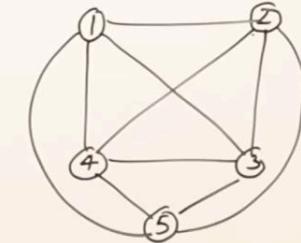
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	2	3	4	5	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

②

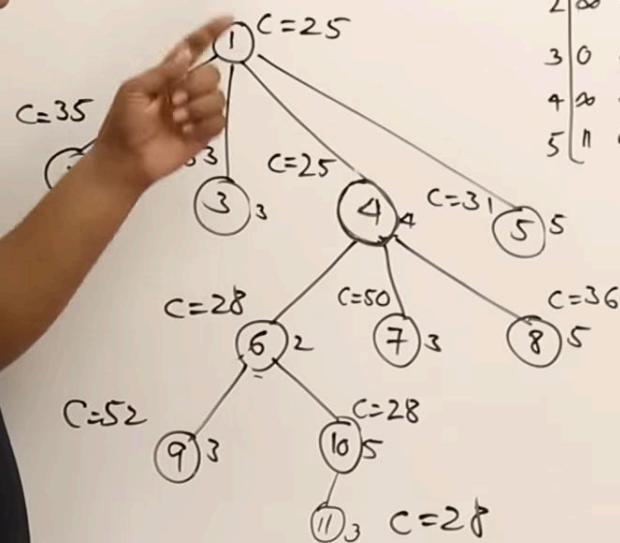
③

④

⑤



upper = ∞



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

⑥

⑦

⑧

Reduced cost = 25

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

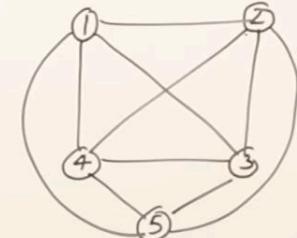
⑨

⑩



Traveling Salesperson Branch-n-Bound

	1	2	3	4	5
1	0	2	3	4	5
2	2	0	0	0	0
3	0	0	0	0	0
4	15	0	12	0	0
5	11	0	0	0	0



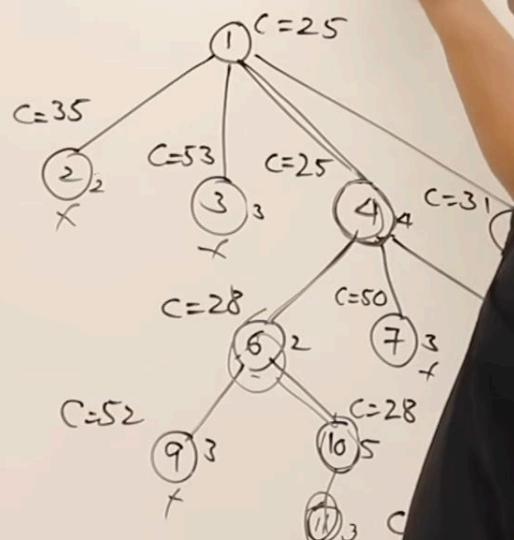
1-4-2-5-3-1

(2)

(4)

(5)

upper = 26



	1	2	3	4	5
1	0	2	3	4	5
2	2	0	0	0	0
3	0	0	0	0	0
4	15	3	12	0	0
5	11	0	0	0	0

reduced cost = 25

	1	2	3	4	5
1	0	2	3	4	5
2	2	0	0	0	0
3	0	20	0	0	0
4	0	20	0	0	20
5	0	20	0	20	0

(10)

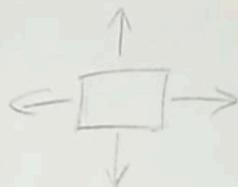


15 Puzzle problem (Using branch and bound)

15 - PUZZLE PROBLEM

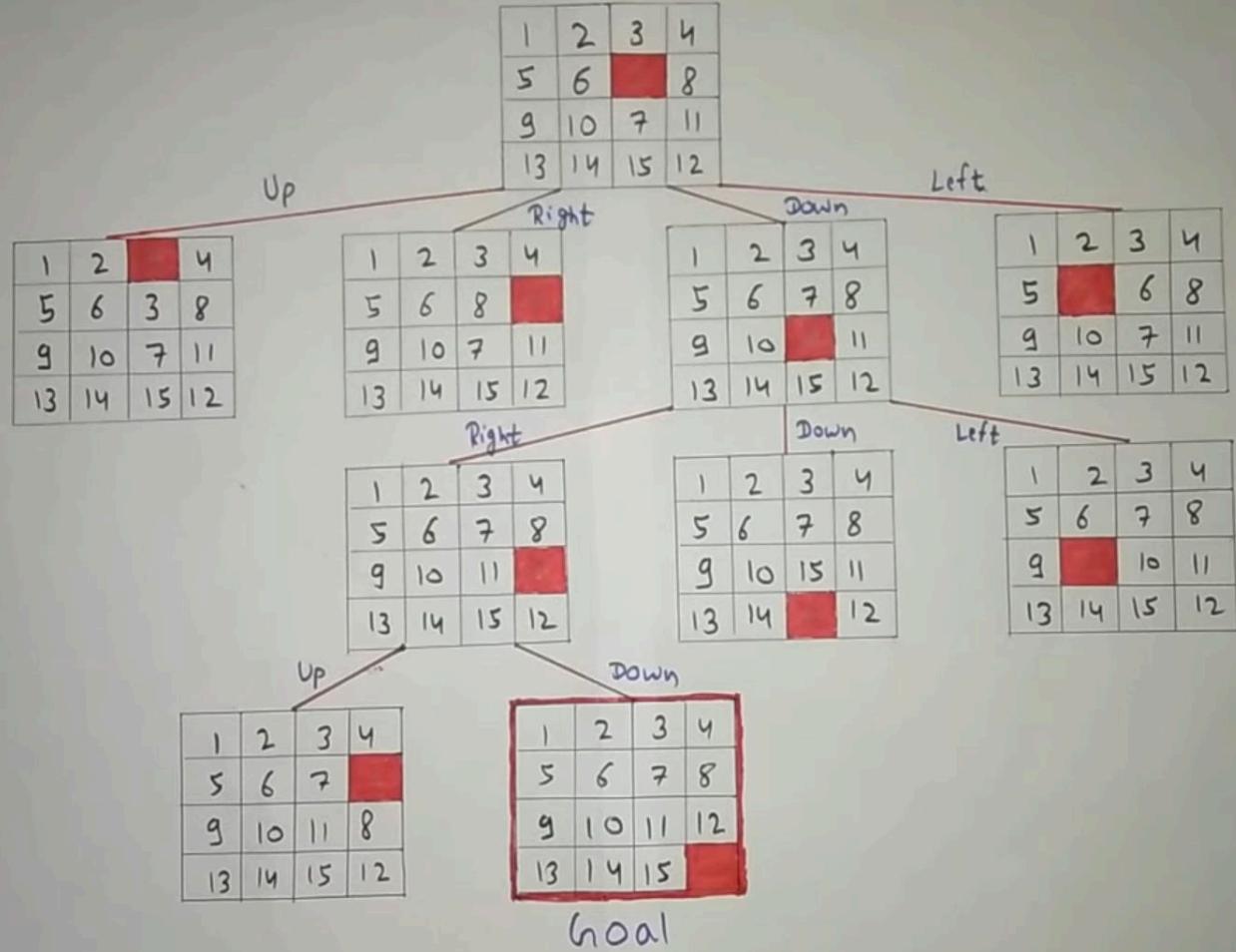
1	2	3	4
5	6		8
9	10	7	11
13	14	15	12

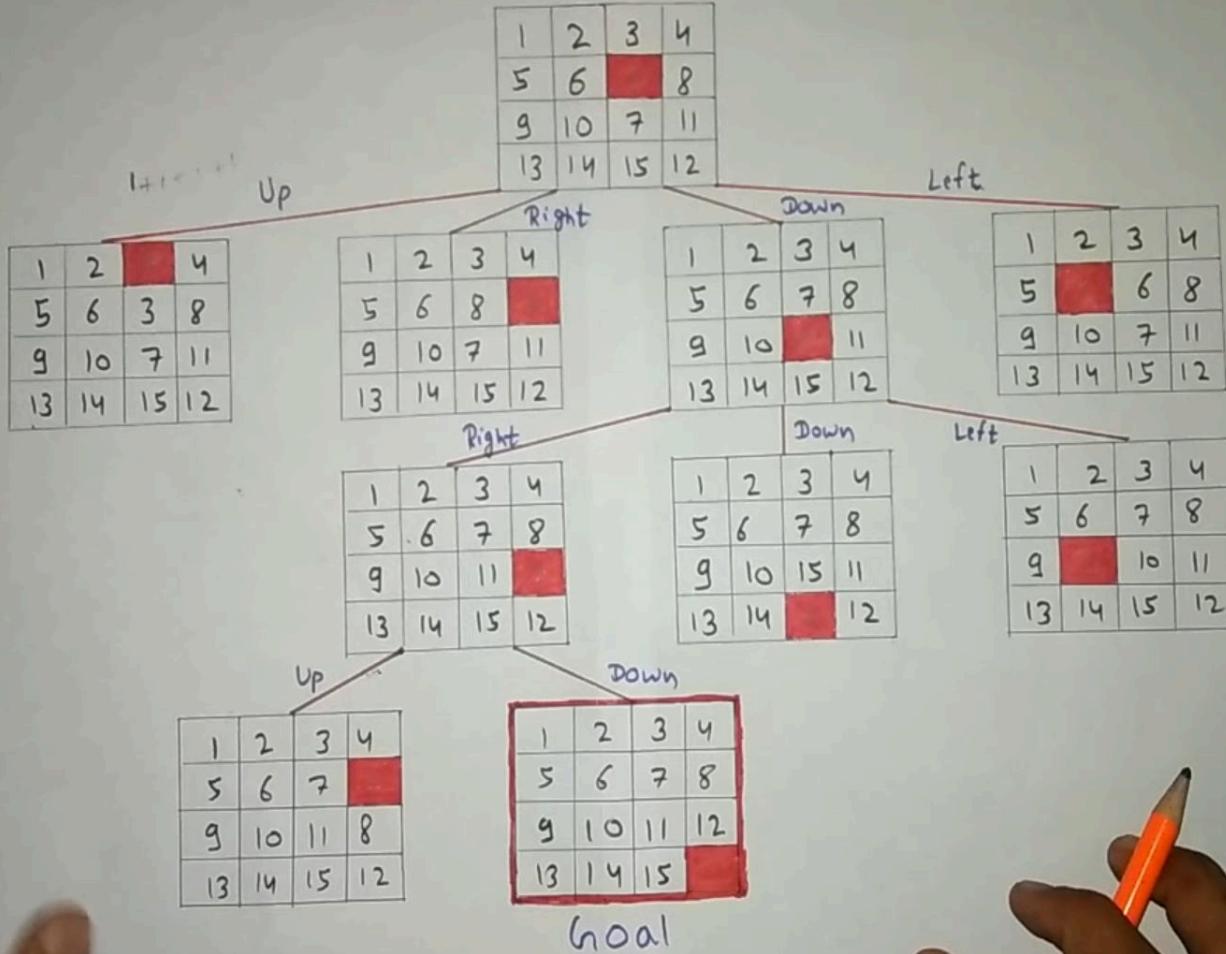
An arrangement

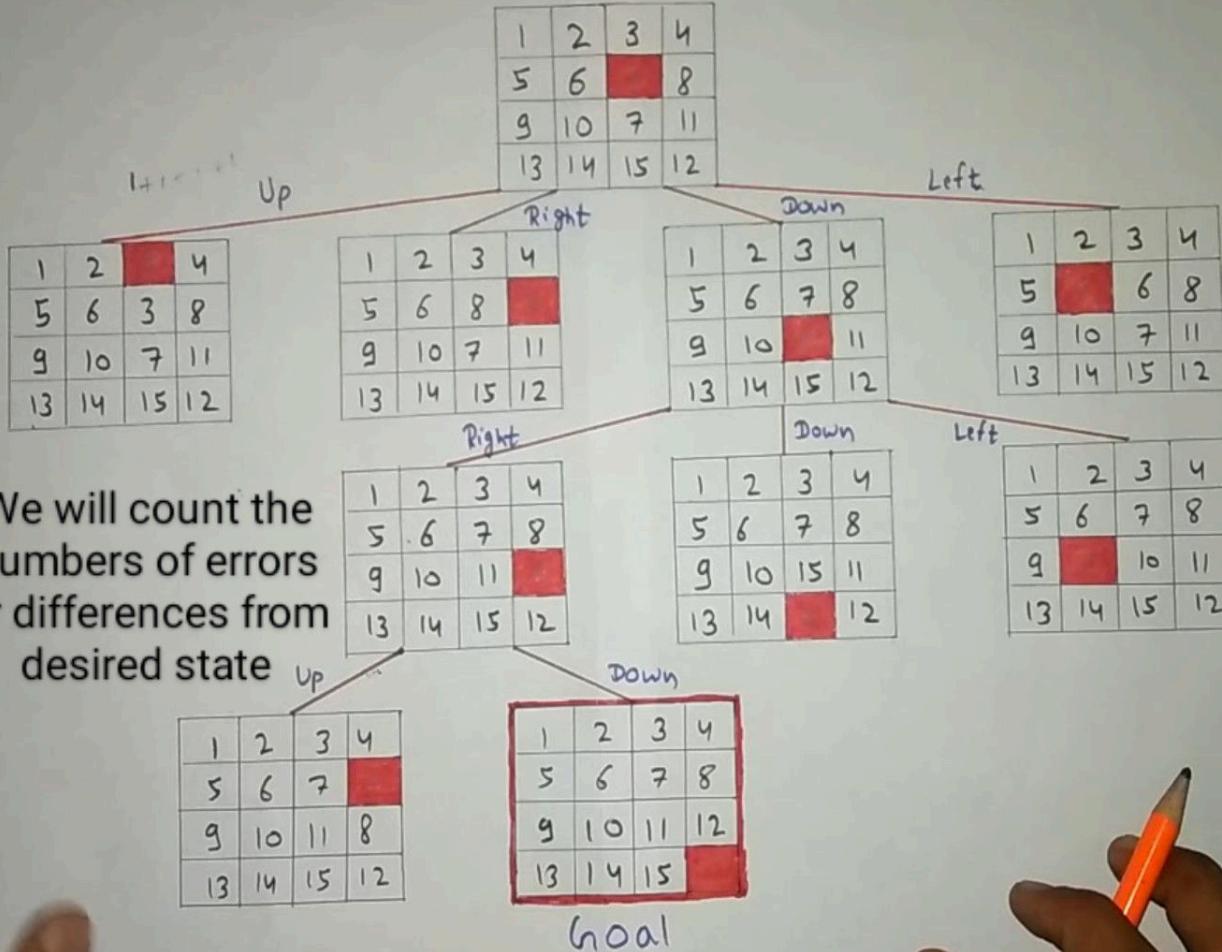


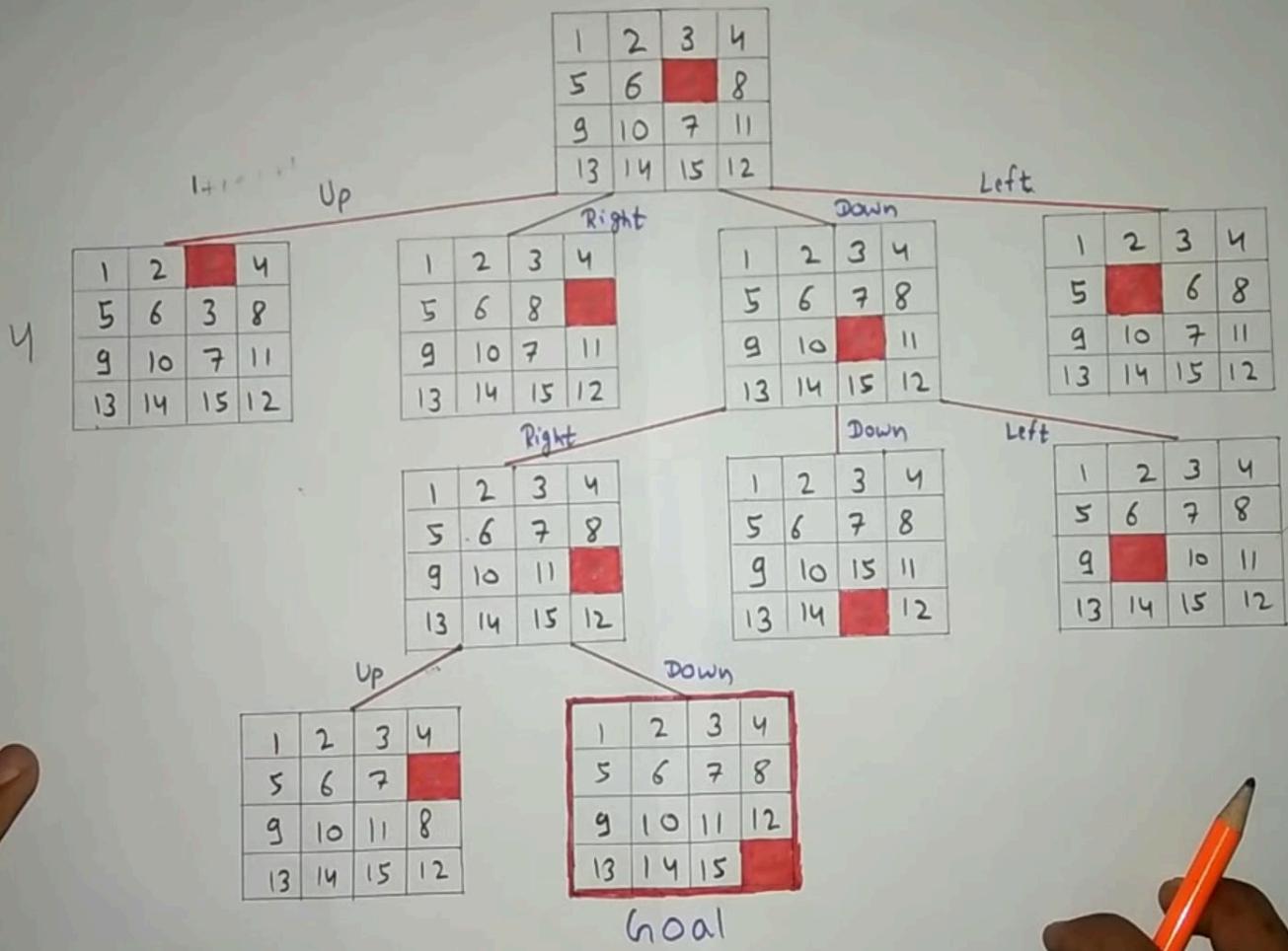
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

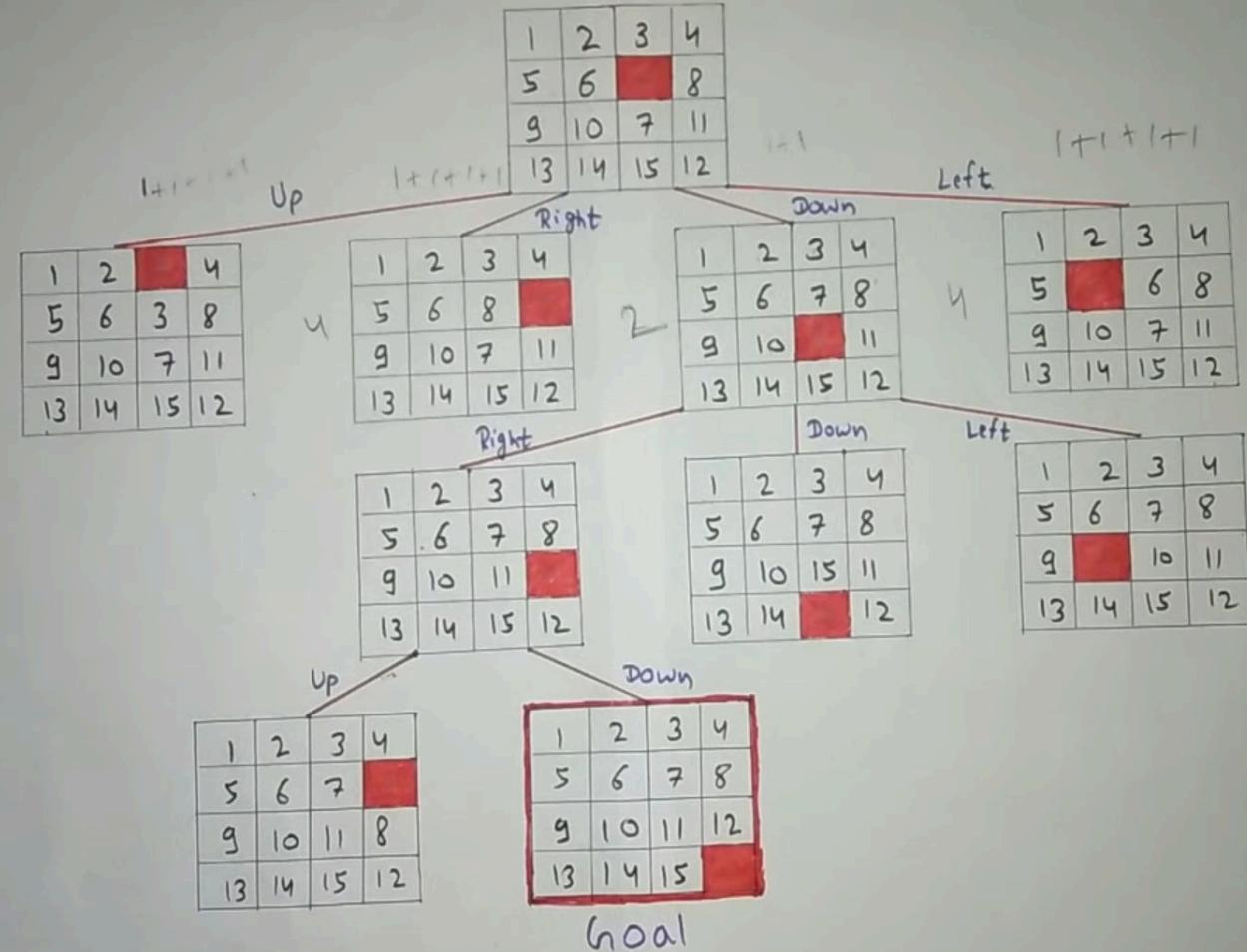
Goal arrangement











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

1+1+1+1
Up

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

1+1+1+1
Right

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

Down

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

Left

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

1+1+1+1
Left

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

1+1+1+1
Right

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

Down

1	2	3	4
5	6	7	8
9	10	11	18
13	14	15	12

1+1
Up

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

Goal