Module - V

Backtracking & Branch and Bound.

A problem has more than 1 solution we use Backtracking.

Branch and Bound technique used to solve optimization problems by constructing state-space-tree. In this technique we obtain optimal solution.

For, Maximization problem we impose upper bound
For, Minimization problem we impose lower bound.

We can consider nodes in the state-space tree as promising and Non promising nodes.

Promising $\xrightarrow{\text{leads to}}$ Solution

Non promising nodes are the ones that do not lead to the solution.

- The nodes bound is not better than other nodes.
- The node violates certain constraints.

Knapsack

$ub = V + (M-W)(V_{i+1}/W_{i+1})$

item	weight	value	value/weight
1	4	40	10
2	7	42	6
3	5	25	5
4	3	12	4

W = total weight of objects which are chosen in the knapsack

V = V is the profit of all chosen objects in the knapsack item $i=0$: $V_1/w_1 = 10$

$w=0$, $v=0$ // No objects are selected.

$M-W = 10-0 = 10$

$ub = 100 = 100$

item $i=1$; $V_2/W_2 = 6$

with item

$W =$ wt of item 1 $+ w = 4$

$V = v + V_1 = 40$

$(M-W) = 10-4 = 6$

$ub = v + (M-W) V_2/W_2 = 40 + 6 * 6 = 76$

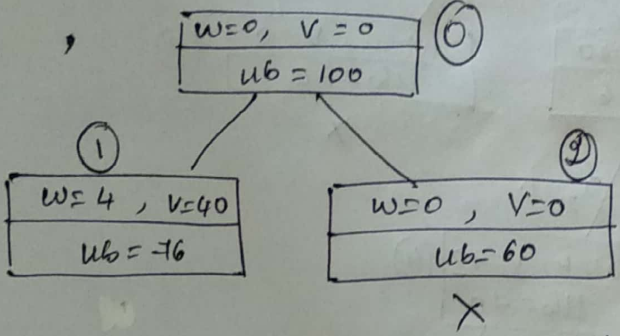
W/O item.

$W = w/o$ item1 $+ w = 0$

$V = v + w/o\ V_1 = 0$

$M-W = 10$

$ub = 60$



In the state-space tree, consider node 1 to expand the tree, As it has max upper bound

item $i=2$ $V_3/W_3 = 5$ (considering node 1)

With item 2

$W =$ wt of item 2 $+ W$
$= 7 + 4 = 11$

Not feasible. Since $W > M$
i,e, $11 > 10$

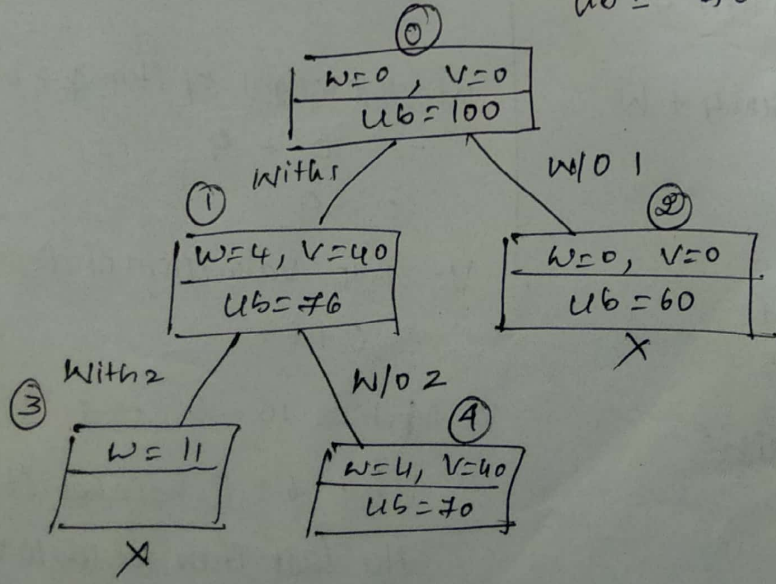
W/O item 2.

$W =$ W/O of item 2 $+ W$
$= 0 + 4 = 4$

$V = 0 + 40 = 40$

$M-W = 10-4 = 6$

$ub = 70$



item $i = 3$, $V_4/W_4 = 4$ [Considering node 4]

With 3

$W = 5 + 4 = 9$

$V = V_4 + V_3 = 40 + 25 = 65$

$M - W = 10 - 9 = 1$

$ub = 69$

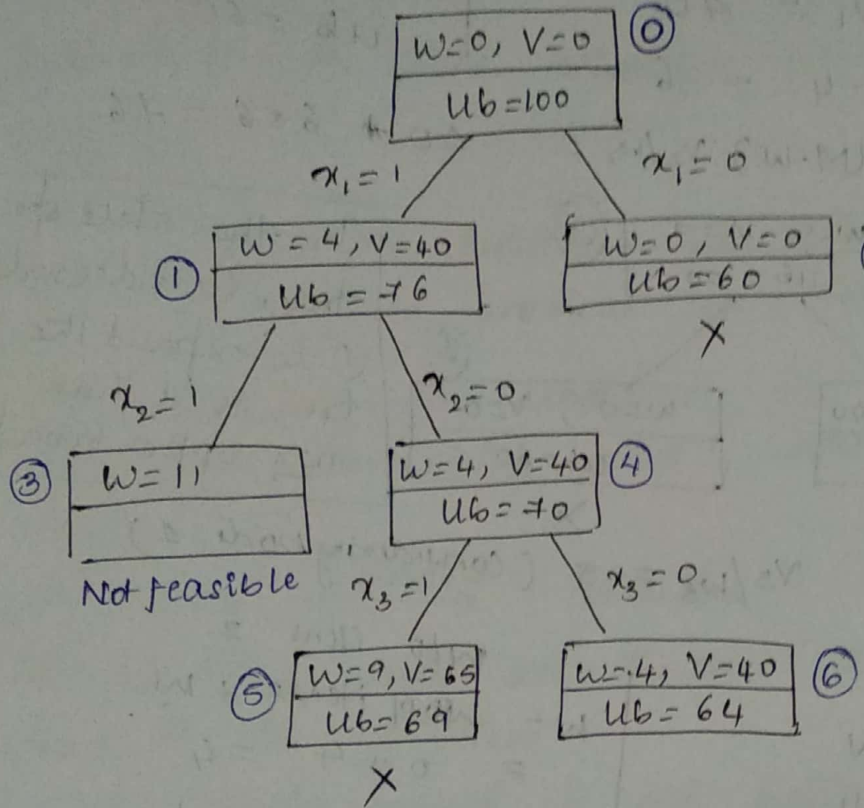
W/O 3

$W = 0 + 4 = 4$

$V = 40$

$M - W = 6$

$ub = 64$



item $i = 4$: V_5/W_5 = ? (Consider node 5 in above figure)

With item 4

W = weight of item 4 + W

$= 3 + 9$

$= 12$

As $W > M$

$12 > 15$

Not feasible

W/O item 4

W = W/O weight of item 4 + W

$= 0 + 9$

$= 9$

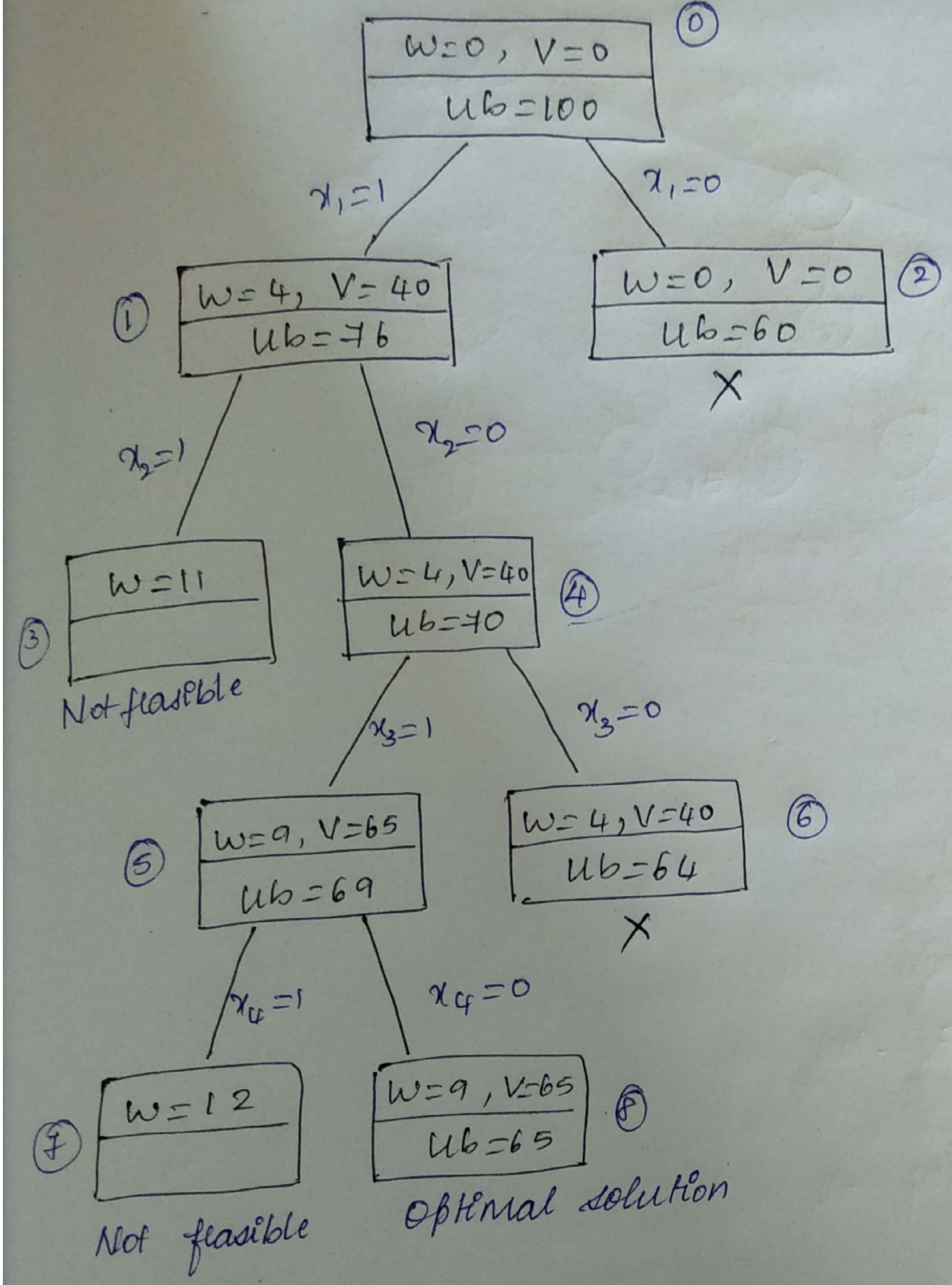
V = W/O value/profit of item 4 + V

$= 0 + 65$

$M - W = 10 - 9 = 1$

$ub = 65$ (because it is the last item set ub to V)

The final state - space tree is,



So final solution = {1, 3} i.e., 1st and 3rd objects are selected.

Max. Profit V = 40 + 25 = 65

Assignment Problem [Minimization]

Definition : Given n jobs < $j_1, j_2, j_3, j_4, \dots j_n$ > and n persons < $p_1, p_2, p_3, p_4 \dots p_n$ >, it is required to assign to all n persons with the constraint that 1 job has to be assigned with 1 person. And the cost involved should be minimum

The solution to this problem can be expressed using a single Dimensional array
< $j_1, j_2, j_3, \dots j_n$ >

→ here $j_1, j_2, j_3 \dots j_n$ represent job no.
→ subscripts 1, 2, 3 ... n represent persons.

Eg: < 2, 3, 4, 1 >

- For 1st person, job 2 is assigned
  - 2nd → 3
  - 3rd → 4
  - 4th → 1

Problem
Find the optimal solution for the given assignment problem which is represented as a matrix as shown below.

Jobs →

Persons ↓	J_1	J_2	J_3	J_4
a	9	2	7	8
b	6	4	3	7
c	5	8	1	8
d	7	6	9	4

The cost involved for ith person to complete jth job is represented using cost adjacency matrix. The exhaustive approach to the assignment problem requires all permutations of integers 1, 2, 3 . . n for n jobs and n people. For each permutation we have to obtain the total cost.

Total no of feasible solutions for $n=4$ will be 24. Which is nothing but $4!$ So in general, if there are n jobs to be assigned to n people then the total number of feasible solutions will be $n!$ Hence the time complexity in the worst case is given by $T(n) \in O(n!)$ here, we observe that the algorithm with time complexity of $O(n!)$ takes too much time and hence it is less efficient.

×

Branch and Bound Solution
for
Assignment problem.

Step 1: Find lower bound for the problem given

Step 2: To get the initial lower bound take minimum of each row and add them.

	J_1	J_2	J_3	J_4	Minimum
a	10	3	8	9	→ 3
b	7	5	4	8	→ 4
c	6	9	2	9	→ 2
d	8	7	10	5	→ 5

lb = 14

Consider person a : and assign various jobs to 'a'

Let a→1 with cost = 10, Leave row a and column 1 and calculate the lower bound

	J_1	J_2	J_3	J_4	Minimum
a	10	3	8	9	→ 10
b	7	5	4	8	→ 4
c	6	9	2	9	→ 2
d	8	7	10	5	→ 5

lb = 21

Let a→2 with cost = 3, Leave row a and column 2 (a is assigned job 2) & calculate lower bound

	J_1	J_2	J_3	J_4	Minimum
a	10	3	8	9	→ 3
b	7	5	4	8	→ 4
c	6	9	2	9	→ 2
d	8	7	10	5	→ 5

lb = 14

Let a → 3 with cost = 8, Leave row a & column 3 & calculate lower bound

	J_1	J_2	J_3	J_4	Minimum
a	10	3	8	9	→ 8
b	7	5	4	8	→ 5
c	6	9	2	9	→ 6
d	8	7	10	5	→ 5

lb = 24

Scanned by CamScanner

Let a ⟶ 4 with cost 9, leave row a & Column 4
(a is assigned job 4)

	J_1	J_2	J_3	J_4	Minimum
a	~~10~~	~~3~~	~~8~~	9	→ 9
b	7	5	4	~~8~~	→ 4
c	6	9	2	~~9~~	→ 2
d	8	9	10	~~5~~	→ 8

lb = 22

In the state space tree, we must consider node ② has it is having least lower bound i.e., assign job 2 to person a with cost = 3.

Consider person b : Assign various ~~jo values~~ to jobs to 'b' [Except job 2, As it is assigned to a]

Let b → 1 with cost 7, Leave row b & column 1
(assign job 1 to b)

Calculate lower bound

	J_1	J_2	J_3	J_4	minimum
a		3			→ 3
b	7		4	8	→ 4
c	6		2	9	→ 2
d	8		10	5	→ 5

lb = 17

Let b → 3 With cost 4

	J_1	J_2	J_3	J_4	→ minimum
a		3			→ 3
b	7	- -	4	8	→ 4
c	6		2	9	→ 6
d	8		10	5	→ 5

lb = 18

, Leave row b and column 3

Let b → 4 with cost 8, Leave row b and column 4

	J_1	J_2	J_3	J_4	Minimum
a		3			→ 3
b	7	- -	4	8	→ 8
c	6		2	9	→ 2
d	8		10	~~5~~	→ 8

lb 21

Consider person c [Assigning job 3 & job 4]
, Let c → 3 With cost 2

	J_1	J_2	J_3	J_4	Minimum
a		3			→ 3
b	7				→ 7
c			2	~~9~~	→ 2
d			~~10~~	~~5~~	→ 5
					17

$\boxed{lb = 17}$

Leave row c and column 3

c → 3
d → 4

Since c is assigned to job 3, job 4 must be assigned to d.

Let C → 4 with cost = 9

	J1	J2	J3	J4	
a		3			→ 3
b	7				→ 7
c	-	-	2	9	→ 9
d			10	5	→ 10

16 = 29

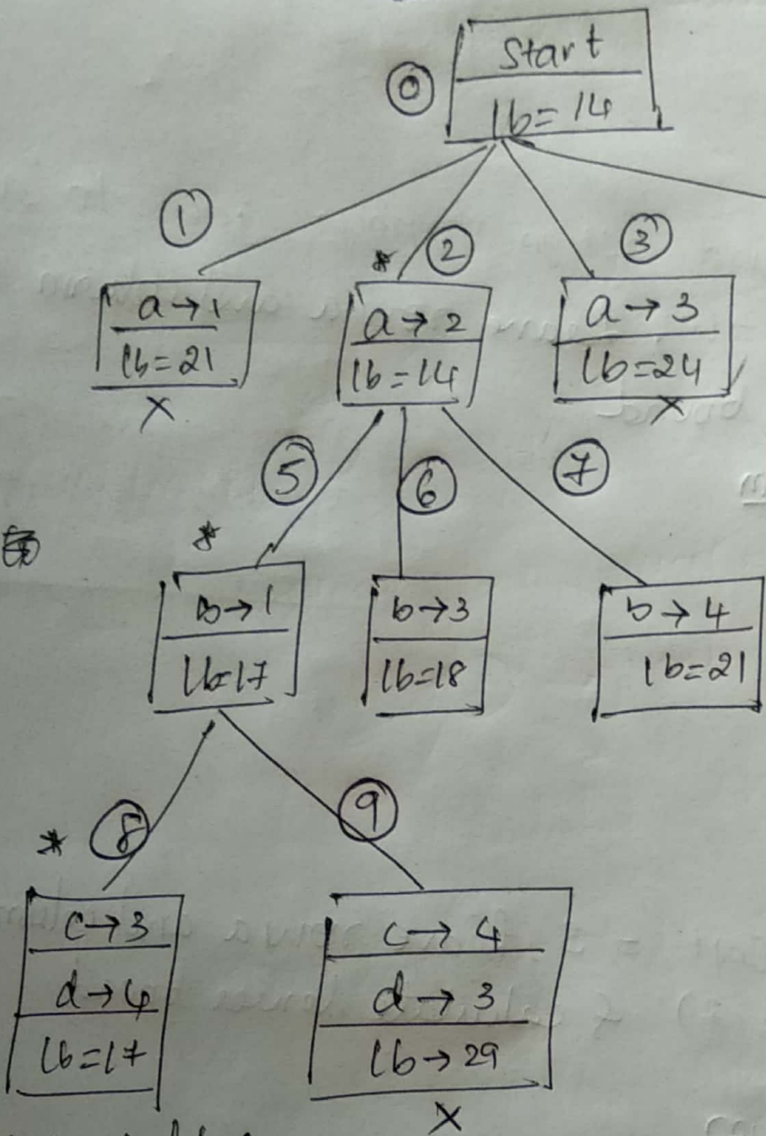
Leave row x and column 4

C → 4

d → 3

Since job 4 is assigned to c

Hence, job 3 is assigned to d



Here optimal solution is denoted by node 8

Optimal Solution

Optimal solution = { a → 2, b → 1 , c → 3, d → 4 }

∴ Cost = 3 + 7 + 2 + 5

∴ $\boxed{\text{Cost} = 17}$ in completing all 4 jobs by 4 persons

Travelling salesman problem
by
Branch - Bound Method

To start with? What is travelling Salesman problem?

Starting from one node we should visit all other (n-1) nodes, exactly once and return to the start.

The main thing is, In what order should we visit the cities to minimize the total distance travelled.

Branch - Break up problem into set of all tours

Bound - lower bound.

AIM: To find least cost tour.

Formula: To calculate $lb = S/2$

lb: lower bound

S: sum of two smallest distance from each city.

Conditions: To reduce the amount of work in constructing state space tree, let us take advantage of following observation.

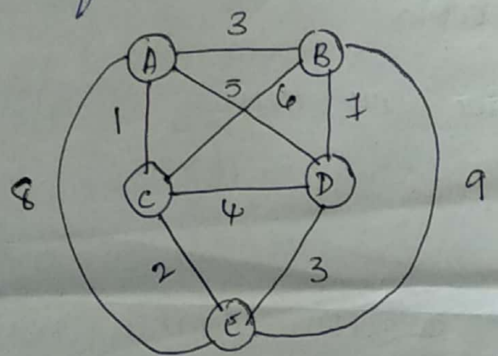
1. Tour always starts at a.
2. The node b is visited before node c
3. First & Last city remains same.

Scanned by CamScanner

Steps

1. Calculate the lowerbound of starting node 'a'
2. Calculate the lowerbound by considering various edges from vertex 'a'.
3. Calculate the lowerbound by considering various edges from vertex 'b'
4. After getting the path calculate the cost

Problem: Apply branch and bound algorithm to solve TSP for the following graph



cost adjacency Matrix :

$$\begin{array}{c|c} & \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \begin{bmatrix} 0 & 3 & 1 & 5 & 8 \\ 3 & 0 & 6 & 7 & 9 \\ 1 & 6 & 0 & 4 & 2 \\ 5 & 7 & 4 & 0 & 3 \\ 8 & 9 & 2 & 3 & 0 \end{bmatrix} \end{array}$$

Step 1 : Calculate the lower bound of starting node.

$$lb = \frac{S}{2} = \frac{\overset{a}{(3+1)} + \overset{b}{(3+6)} + \overset{c}{(1+2)} + \overset{d}{(4+3)} + \overset{e}{(2+3)}}{2}$$

$$= \frac{28}{2} = 14$$

0 $\boxed{\begin{array}{c} a \\ \hline lb=14 \end{array}}$

From here we have some many possibilities i.e., From

$$\left.\begin{array}{l} a \xrightarrow{\text{1 can go to}} b \\ a \longrightarrow c \\ a \longrightarrow d \\ a \longrightarrow e \end{array}\right\} \text{calculate lb.}$$

Where to go ? Based on what should we select

Step2: Calculate the lowerbound for various edges of a.

for Edge (a,b) = $s = \left(\overset{a}{(\underset{(a,b)}{3}+1)} + \overset{b}{(\underset{(b,a)}{3}+6)} + \overset{c}{(1+2)} + \overset{d}{(4+3)} + \overset{e}{(2+3)} \right) / 2$

$lb = 14$

For Edge (a, c) = Not feasible because b is not before c

For Edge (a, e) = $s = \left(\overset{a}{(\underset{(a,e)}{8}+1)} + \overset{b}{(3+6)} + \overset{c}{(1+2)} + \overset{d}{(8+2)} + \overset{e}{(\underset{(e,a)}{8}+2)} \right) / 2$

$= 38/2 = 19$

For Edge (a, d) = $s = \left(\overset{a}{(\underset{(a,d)}{5}+1)} + \overset{b}{(3+6)} + \overset{c}{(1+2)} + \overset{d}{(\underset{(d,a)}{5}+3)} + \overset{e}{(2+3)} \right) / 2$

$= 31/2 = 15.5 \approx 16$

Step3: Calculate the lowerbound for various Edges of b

for Edge (b, c) = $s = \left(\overset{a}{(3+1)} + \overset{b}{(\underset{(b,c)}{6}+3)} + \overset{c}{(\underset{(c,b)}{6}+1)} + \overset{d}{(3+4)} + \overset{e}{(2+3)} \right) / 2$

$= 32/2 = 16$

For Edge (b, d) = $s = \left(\overset{a}{(1+3)} + \overset{b}{(\underset{(b,d)}{7}+3)} + \overset{c}{(1+2)} + \overset{d}{(\underset{(d,b)}{7}+3)} + \overset{e}{(2+3)} \right) / 2$

$= 32/2 = 16$

For Edge (b, e) = $s = \left(\overset{a}{(1+3)} + \overset{b}{(\underset{(b,e)}{9}+3)} + \overset{c}{(1+2)} + \overset{d}{(3+4)} + \overset{e}{(\underset{(e,b)}{9}+2)} \right) / 2$

$= 37/2 = 18.5 \approx 19$

cost = 16

Now. Considering the various edges from c, we have the following paths.

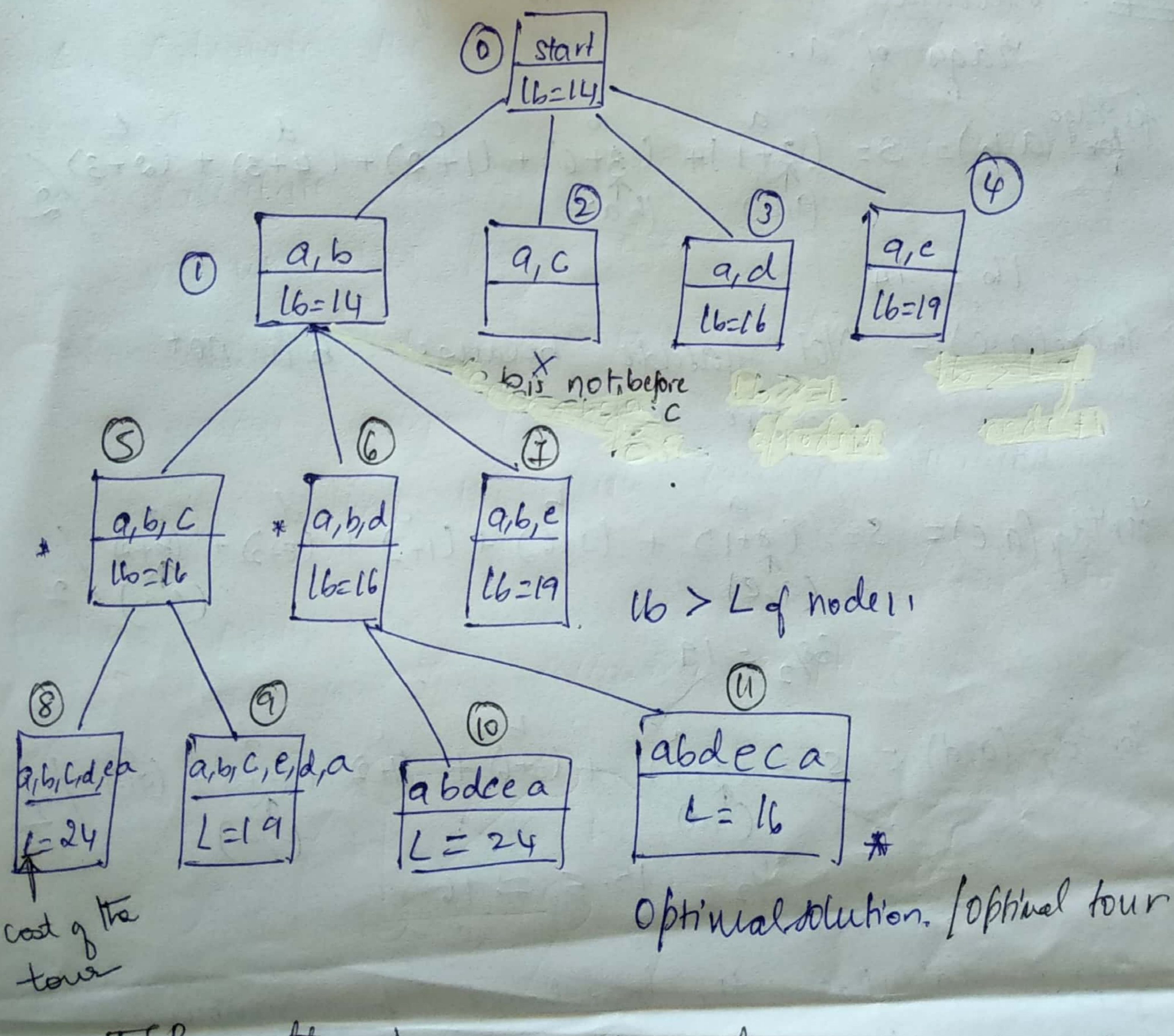
$a \xrightarrow{3} b \xrightarrow{6} c \xrightarrow{4} \boxed{d} \xrightarrow{3} \boxed{e} \xrightarrow{8} a \quad = \quad 24 = \text{Cost}$

$a \xrightarrow{3} b \xrightarrow{6} c \xrightarrow{\cancel{4}} \boxed{e} \xrightarrow{3} \boxed{d} \xrightarrow{5} a \quad = \quad 19 = \text{cost}$

Now, Considering the various edges from d, we have the following paths.

$a \xrightarrow{3} b \xrightarrow{7} d \xrightarrow{4} \boxed{c} \xrightarrow{2} \boxed{e} \xrightarrow{8} a \quad = \quad 24$

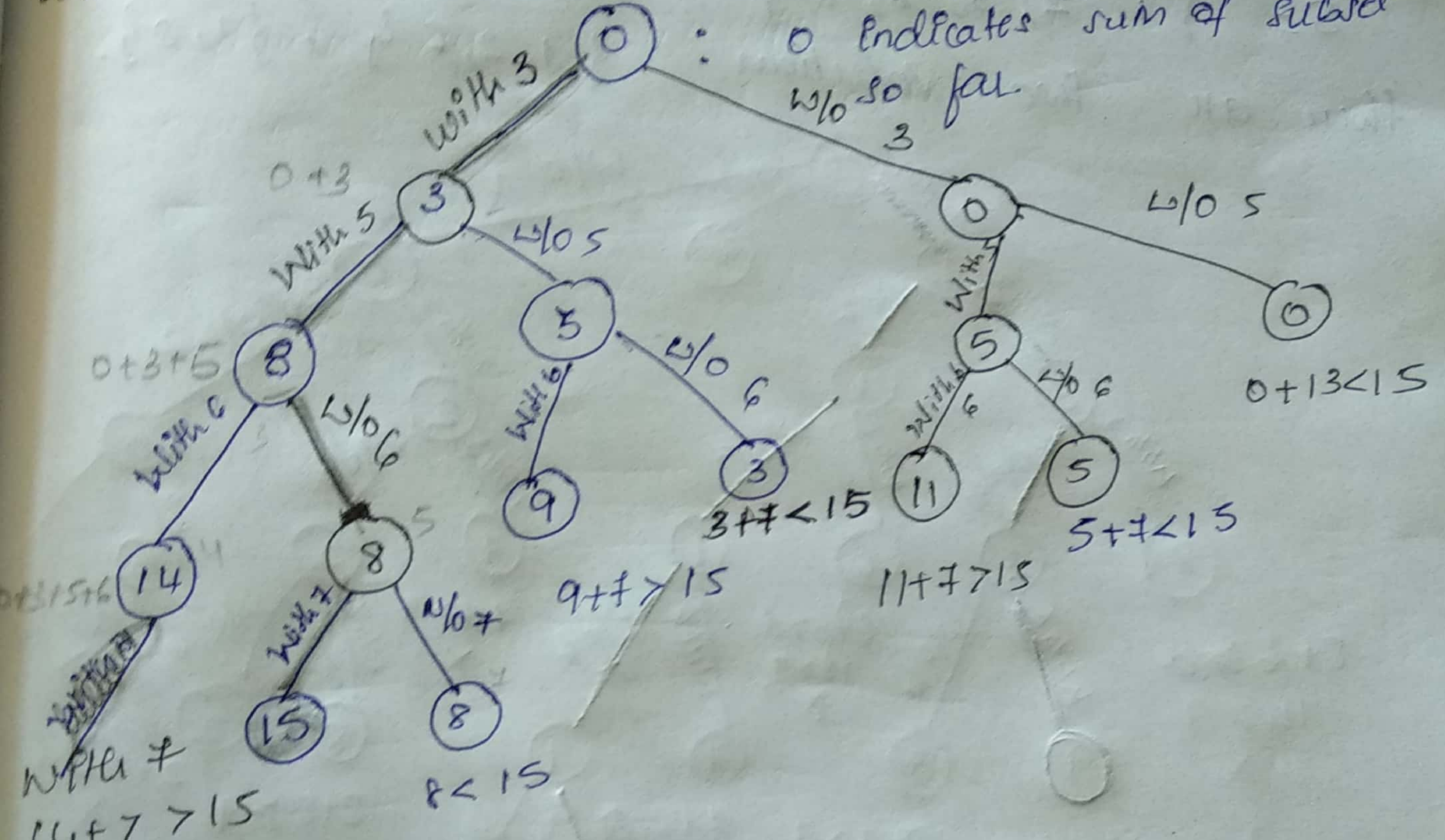
$a \xrightarrow{3} b \xrightarrow{7} d \xrightarrow{3} \boxed{e} \xrightarrow{2} \boxed{c} \xrightarrow{1} a \quad = \quad 16$



TSP path is $a \xrightarrow[3]{} b \xrightarrow[7]{} d \xrightarrow[3]{} e \xrightarrow[2]{} c \xrightarrow[1]{} a$

cost = 16

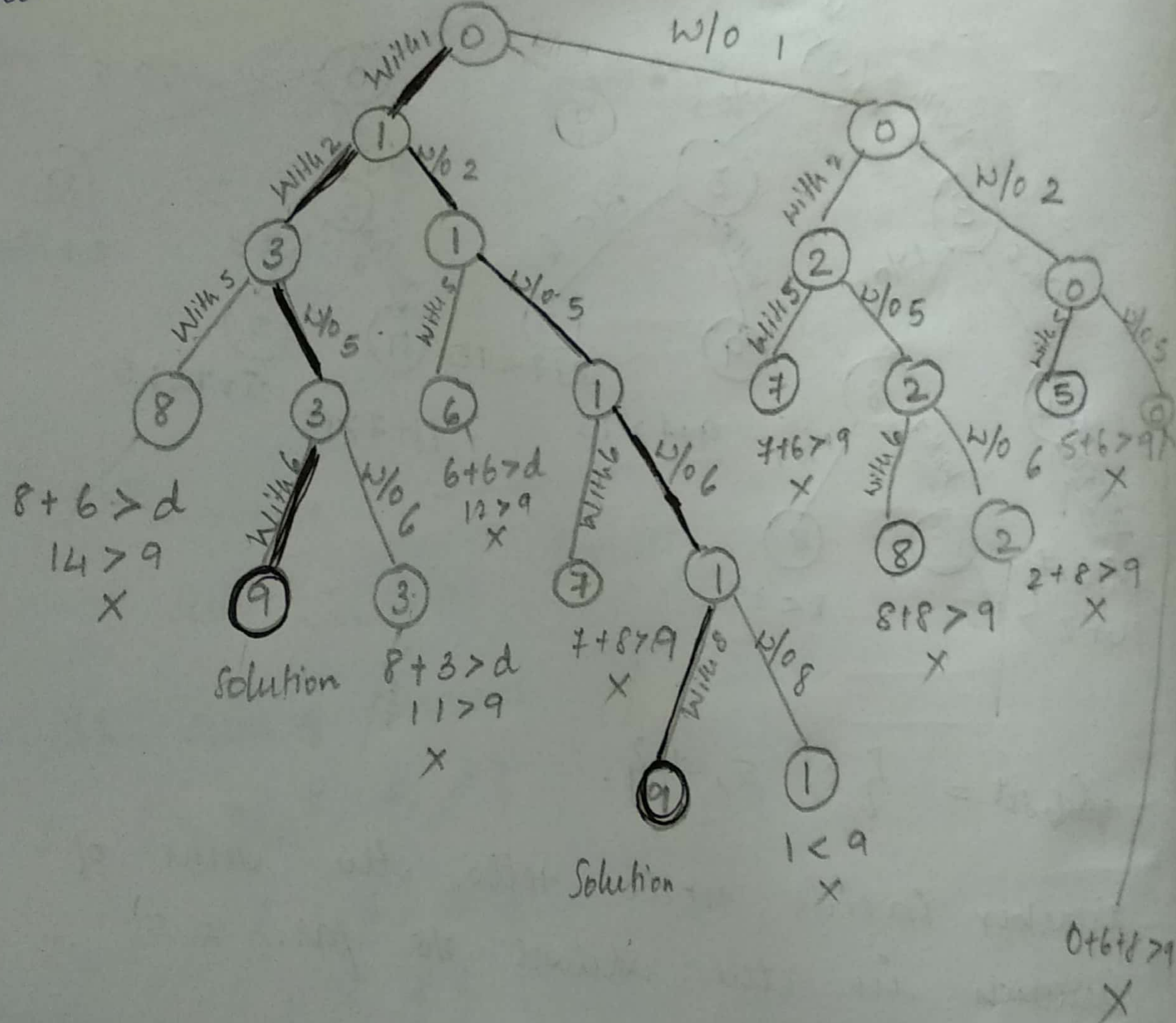
For, $S=\{3, 5, 6, 7\}$ Sub-set - sum Problem
and $d = 15$ solve the sub-set sum problem



Subset = $\{3, 5, 7\}$.

Number inside node tells the sum of elements in the subset so far. = s'

1. Draw state space tree
$S = \{1, 2, 5, 6, 8\}$ and $d = 9$
there are two solutions $\{1, 2, 6\}$, $\{1, 8\}$.



Therefore, solutions are the following subsets obtained from state space tree diagram

$\{1, 2, 6\}$, $\{1, 8\}$