⇒ Back Tracking and Branching :-

They construct candidate solutions one component at a time and evaluate the partially constructed solutions: if no potential values of the remaining components can lead to a solution, the remainining components are not generated at all.

⇒ State - Space Tree :-

* Both the techniques are based on construction of these trees.
* The Nodes reflect specific choices made for a solution's components.
* They terminate a node when it can be gauranteed that no solution to the problem can be obtained by considering choices that correspond to node's descendents.

Brand and Bound	Back Tracking
* Only applicable to Optimization Problems	* More often, applied to non optimization problems.
* Order of generation of nodes is according to best first rule	* Order of generation of nodes is depth first.
Ex:- Knapsack, Job assignment, Travelling Salesman	Ex:- N Queens, Hamiltonian Subset, Subset Sum.

⇒ Back Tracking :-

In State - Space Tree :-

- Root :- Initial State before search
- First level nodes :- Choices made for first component of a solution.
- Promising nodes :- ~~Partially~~ Nodes corresponding to a partially constructed solution that may still lead to a complete solution.
- Non Promising :- Nodes do not lead to a complete solution.
- Leaves :- Either Non - Promising dead ends or complete solutions.

① ⇒ N - Queens Problem :-

The Problem is to place N Queens on a chess board of $n \times n$ squares such that no 2 queens are in same row, column or diagonal.

$n = 1$

Q_1

$n = 2$

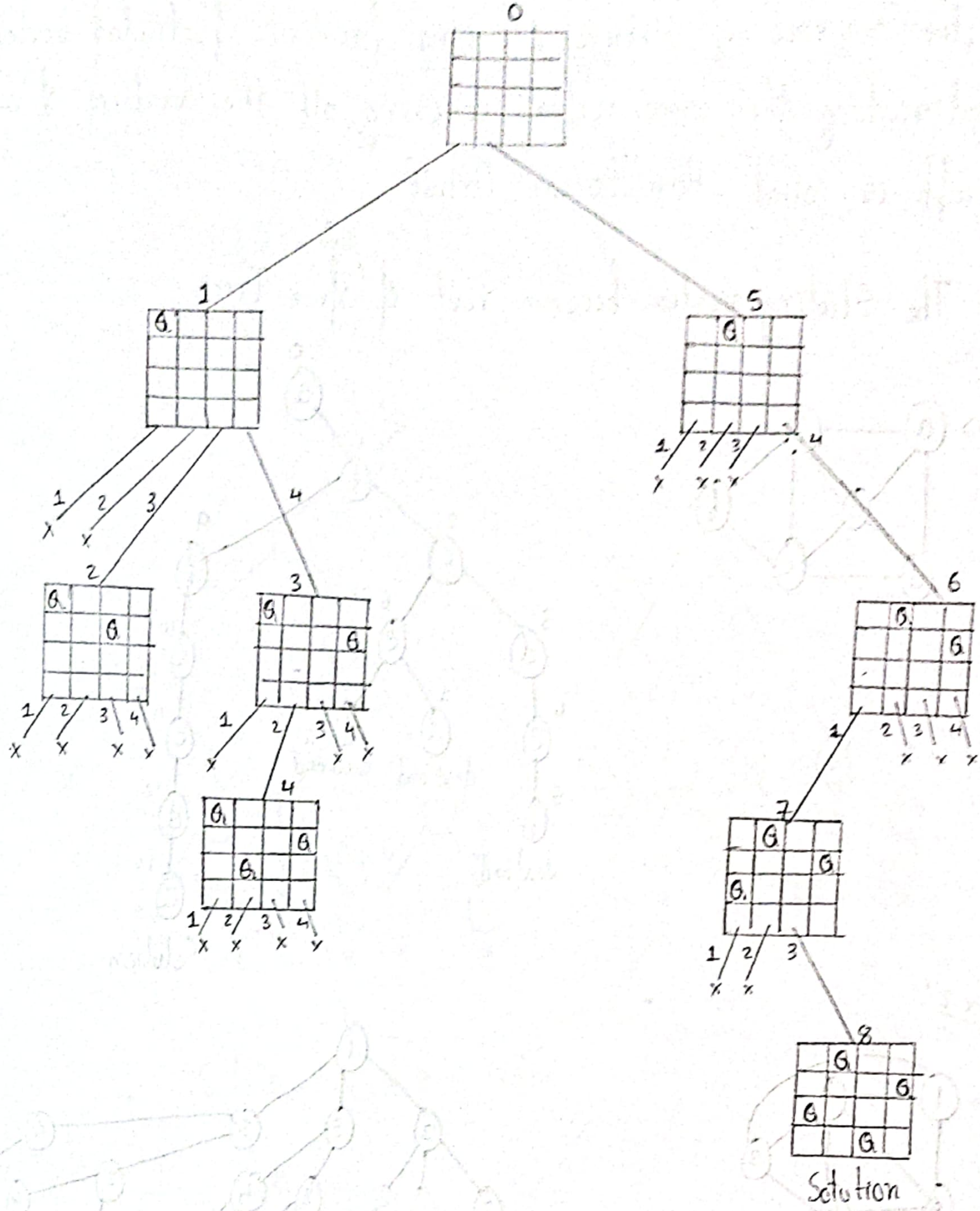
Q_1	

$n = 3$

Q_1		
		Q_2

No Solution for these two cases ($n = 2, 3$)

For $n = 4$:-

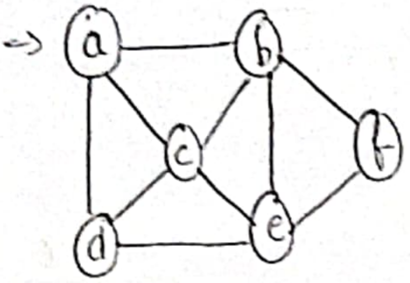


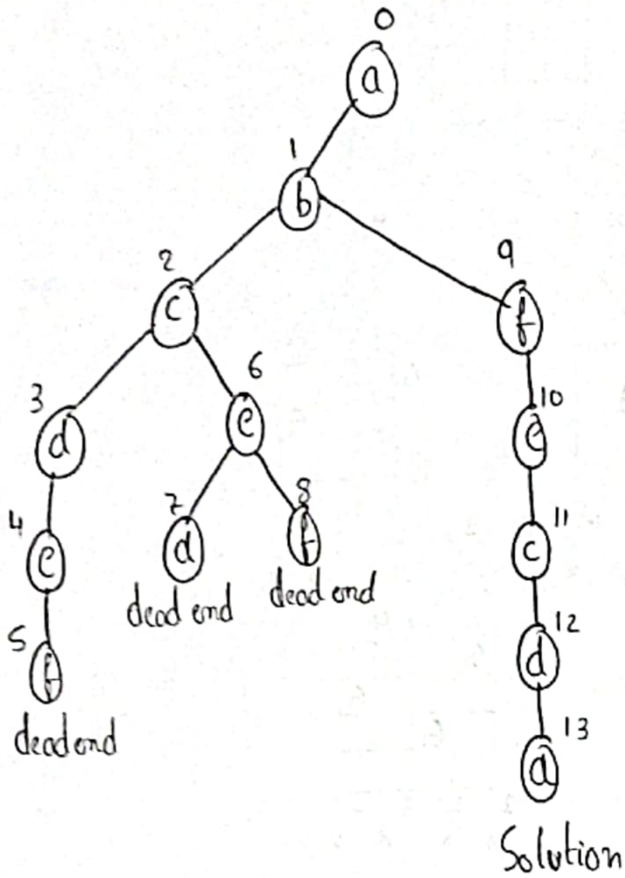
② Hamiltonian Circuit Problem :-

⊗ The sequence of vertices starting from a particular vertex and reaching the same vertex covering all the vertices of a graph is called Hamiltonian Circuit.

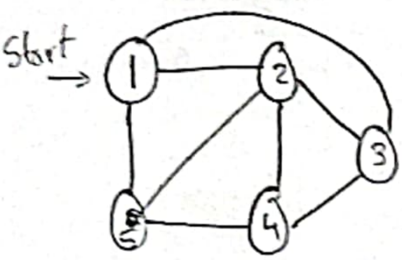
⊗ The Starting vertex becomes root of state space Tree.

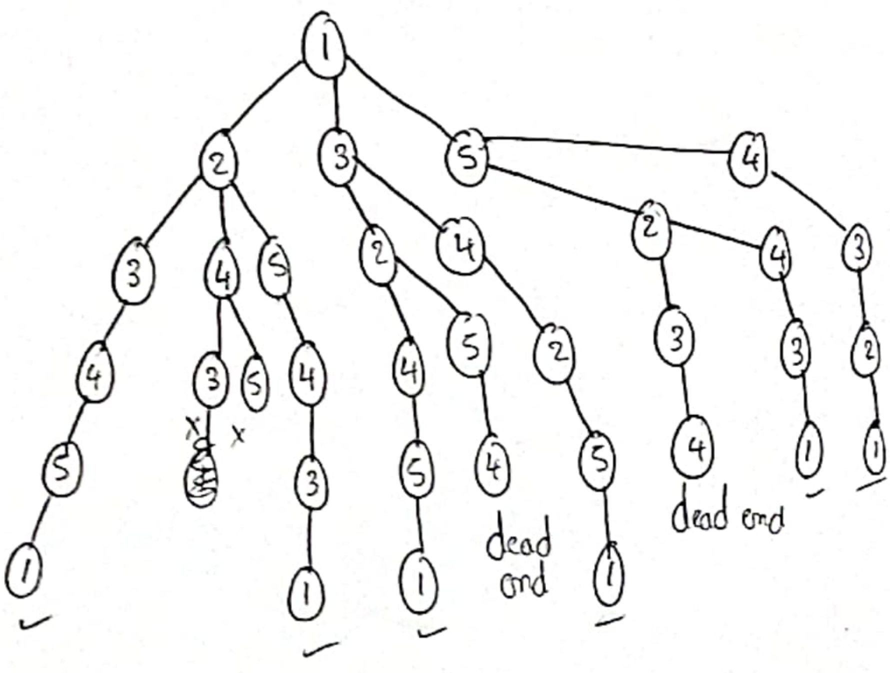
Ex ①:-





Ex 2:-

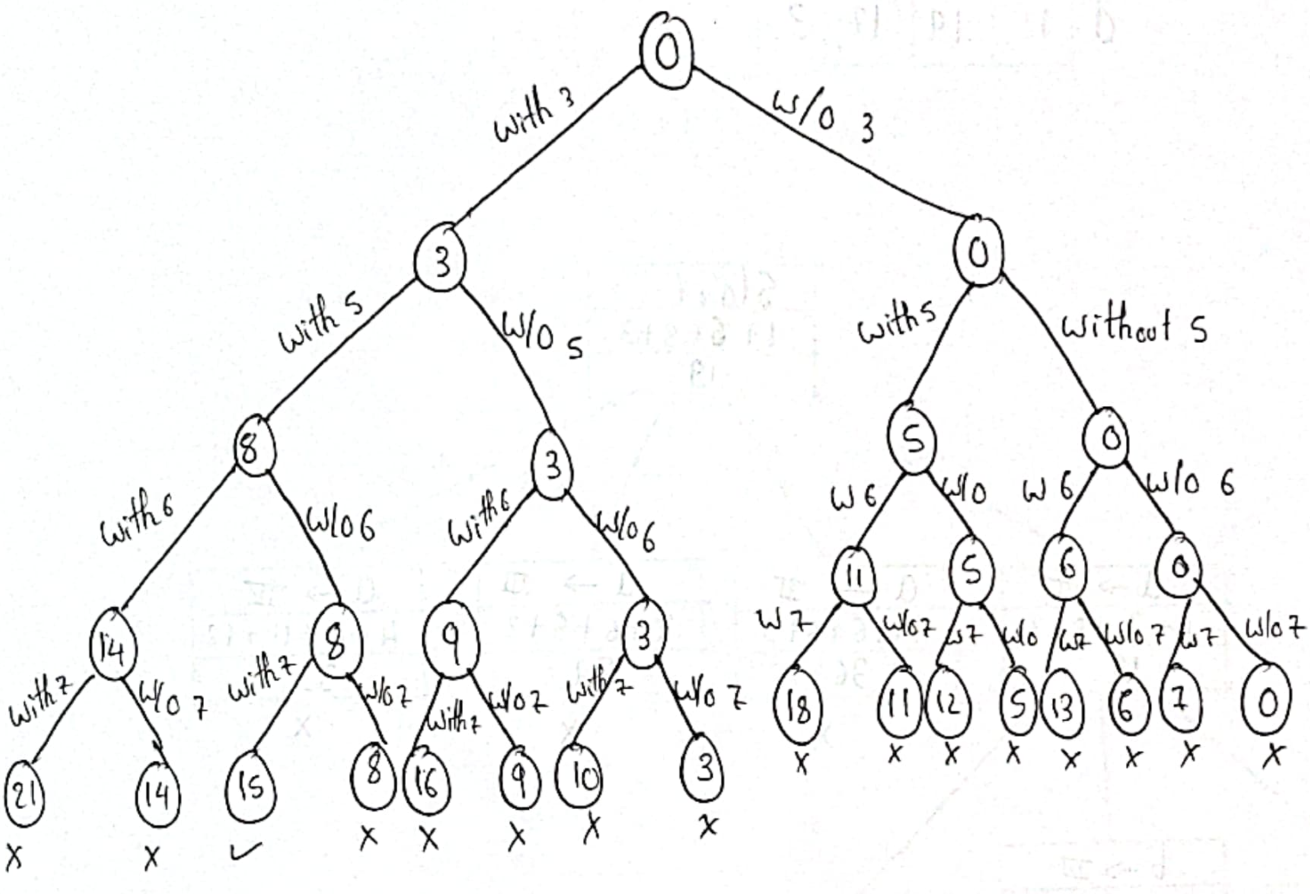




(3) ⇒ Subset-Sum Problem :-

To find a subset of a given set $A = \{a_1, a_2 \dots a_n\}$ for n positive integers whose sum is equal to given positive integer d.

Ex :- $A = \{3, 5, 6, 7\}$ $d = 15$



∴ Solution = $\{3, 5, 7\}$

Note :- Terminating Conditions :-

$s + a_{i+1} > d$ (The Sum s is too large)

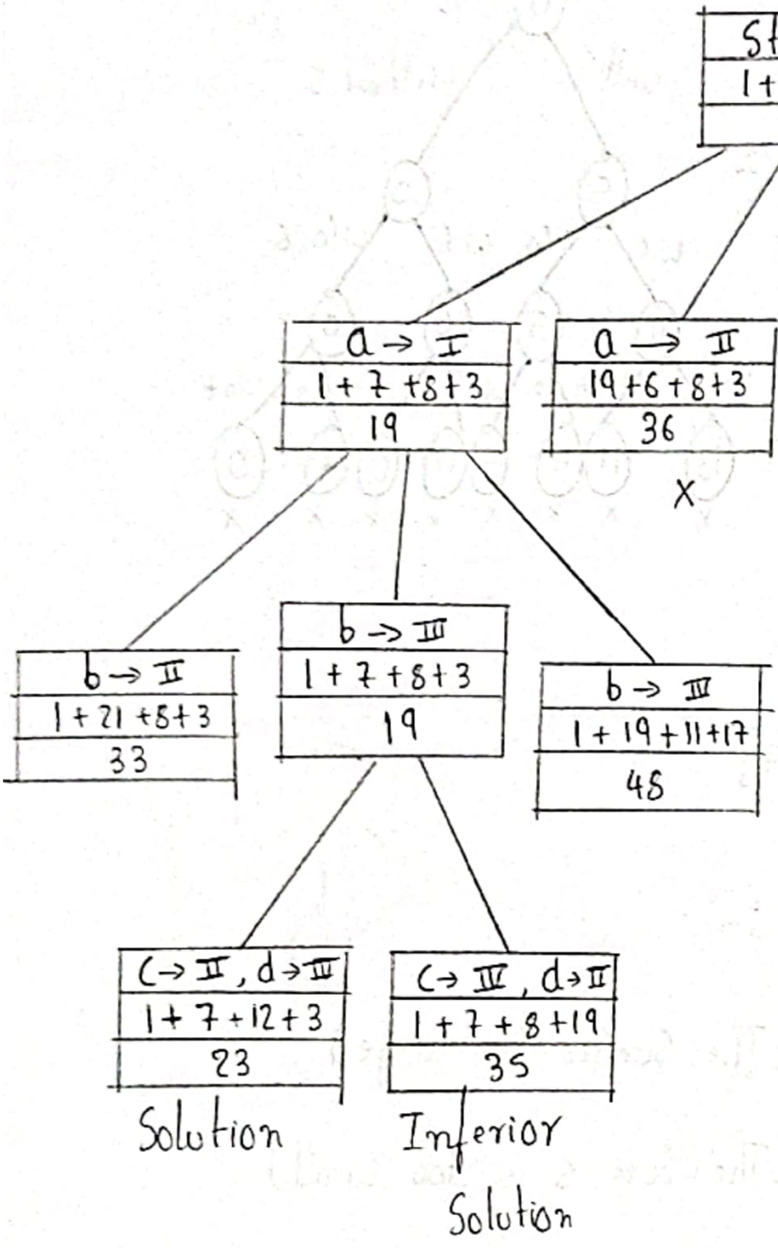
$s + \sum_{j=i+1}^{n} a_j < d$ (The Sum s is too small)

⇒ Branch and Bound :-

① → Job Assignment Problem :-

Ex :-

	I	II	III	IV
a	1	19	10	4
b	6	21	7	19
c	31	12	11	8
d	12	19	17	3



∴ Solution is

a → I, b → III, c → II, d → IV

Cost :- 23

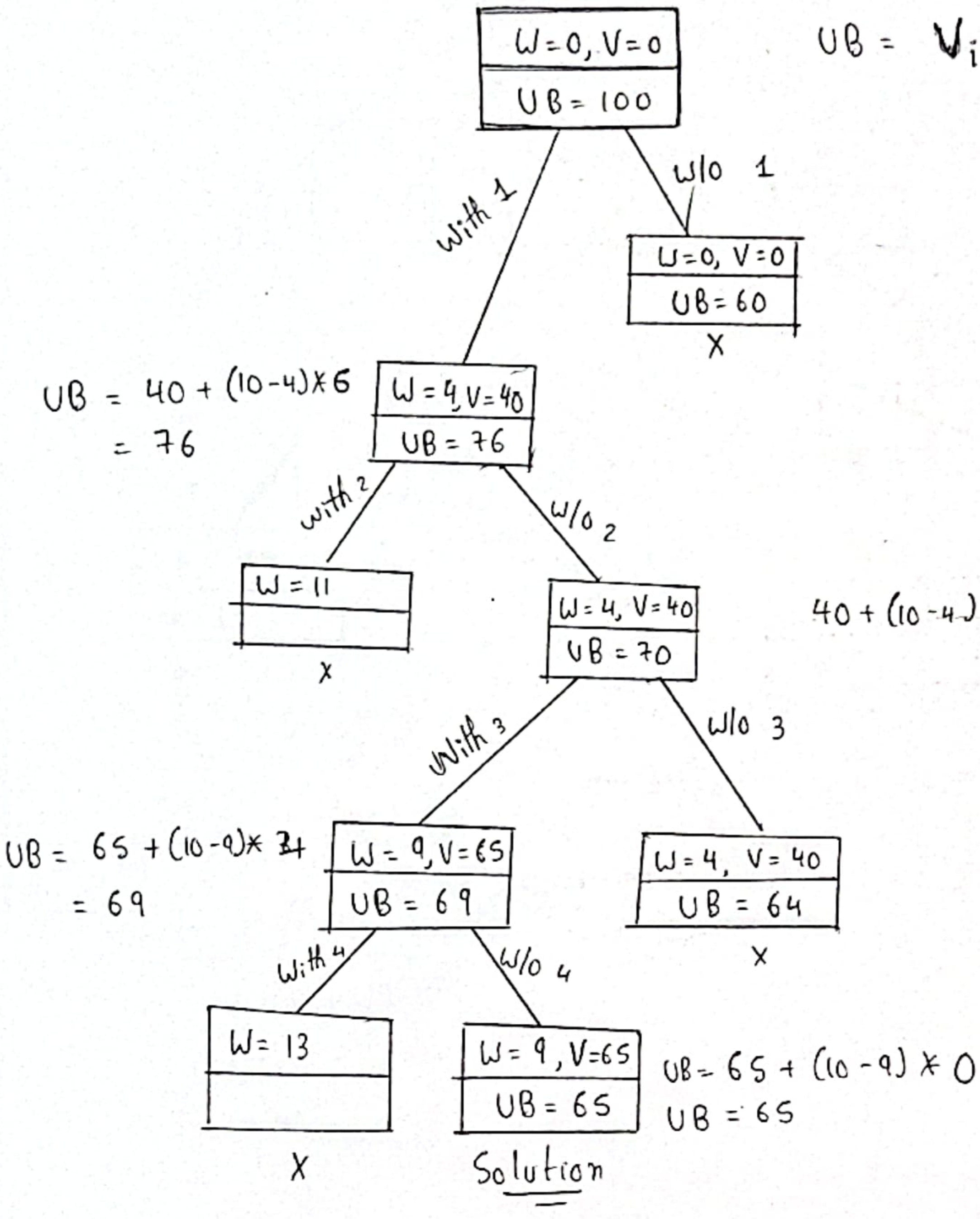
② Knapsack Problem :-

Item	Weight	Value	Value / Weight
1	4	\$40	10
2	7	\$42	6
3	5	\$25	5
4	3	\$12	4

Arrange in descending order of Value/weight ratio

$W = 10$

$UB = V_i + (W - W_i) * (V_{i+1} / W_{i+1})$



Subset = {1, 3}

∴ 1010 Knapsack.

Ex 2 :- $W = 15$ Solution :- 89

Item	Weight	Value	Value/Weight
1	5	40	8
2	7	35	5
3	2	12	6
4	4	4	1
5	5	10	2
6	1	2	2

$\Rightarrow$

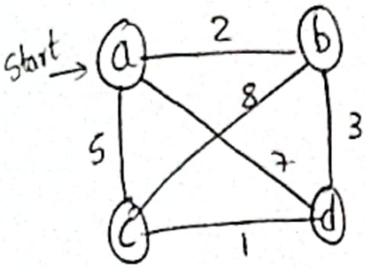
Item	Weight	Value	Value/Weight
1	5	40	8
3	2	12	6
2	7	35	5
5	5	10	2
6	1	2	2
4	4	4	1

Ans :- {1, 3, 2, 6} 111001 Knapsack.

③ Travelling Salesman Problem :-

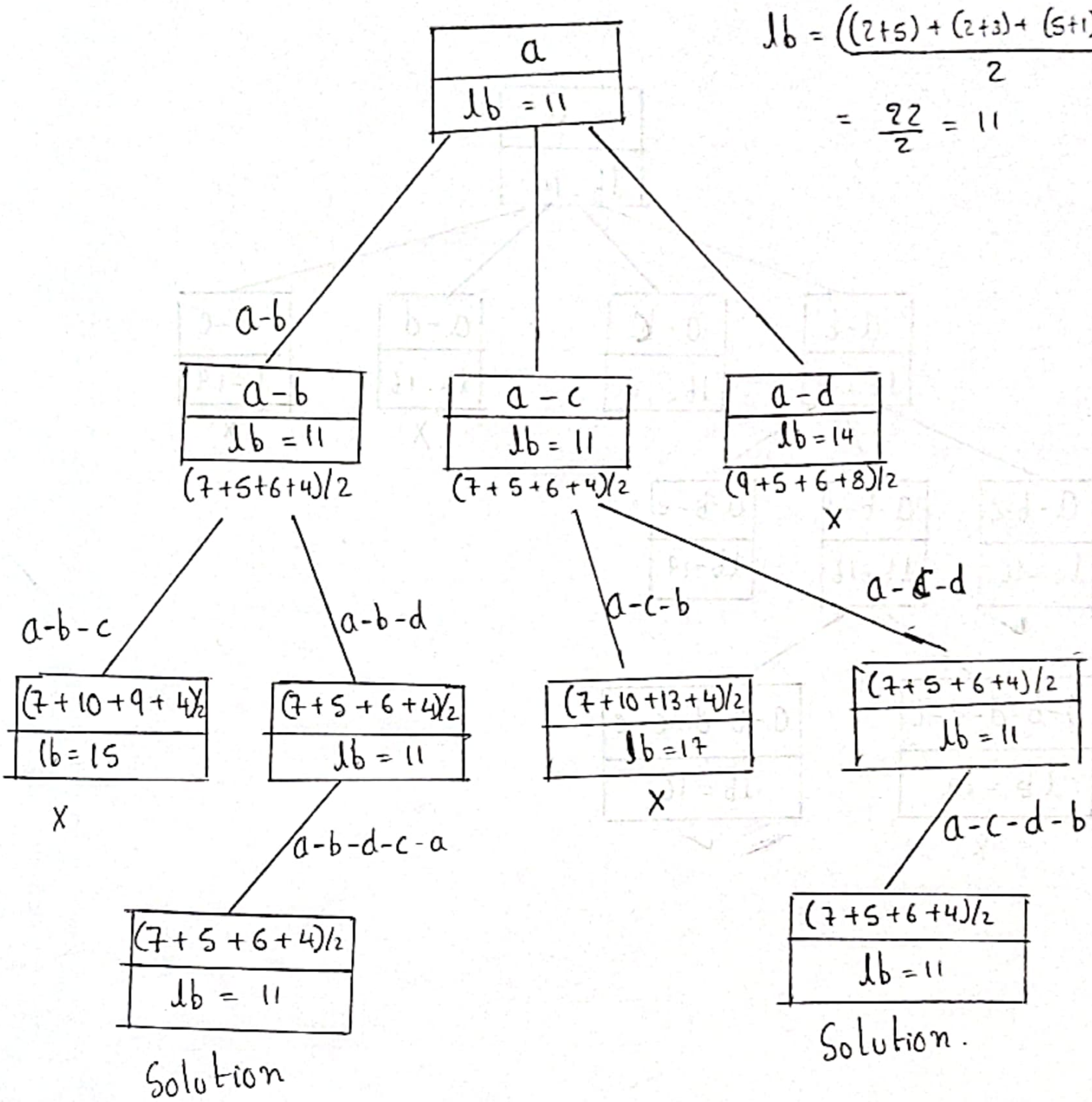
Ex ① :-

Lower Bound = $\dfrac{(\text{Sum of 2 least cost edges of each vertex})}{2}$

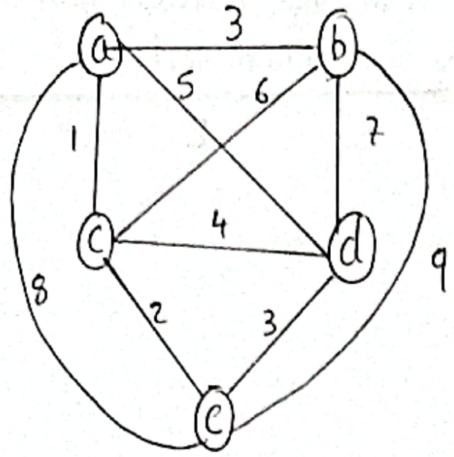


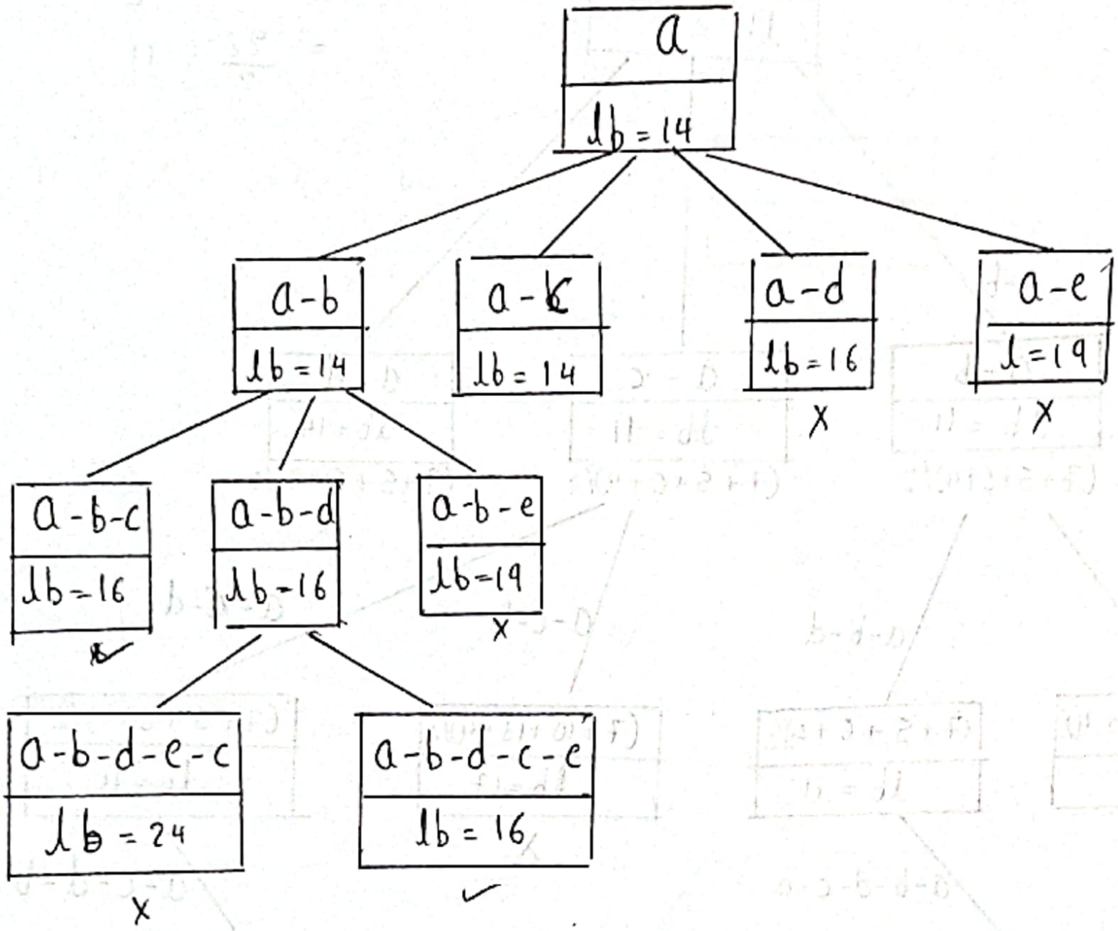
$lb = \dfrac{((2+5)+(2+3)+(5+1)+(1+3))}{2}$

$= \dfrac{22}{2} = 11$



Ex-(2) :-


a
b
c
d
e
3
5
6
1
7
4
8
9
2
3


a
lb = 14
a-b
lb = 14
a-c
lb = 14
a-d
lb = 16
X
a-e
l = 19
X
a-b-c
lb = 16
a-b-d
lb = 16
a-b-e
lb = 19
X
a-b-d-e-c
lb = 24
X
a-b-d-c-e
lb = 16