Assignment - 02

DAA (18CS42)

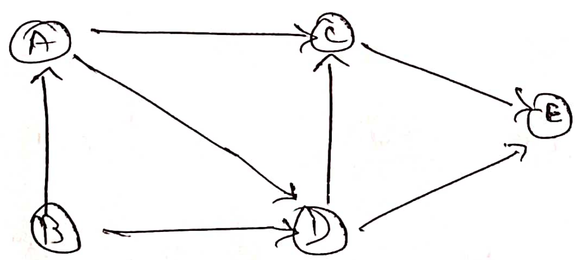
Shahul Hameed . S
1KN18CS097
4th sem CSE 'A' sec

(1) Explain topological sorting with example.

> A directed acyclic graph is a directed graph with no cycle. Based on the principle of DAG. Specific ordering a vertices is possible. This method of arranging the vertices in some specific manner called topological sort. two types of topological sort.
(a) DFS based algorithm. (b) Source Removal algorithm.

(a) DFS Based algorithm: topological sort is a process of assigning a linear ordering to the vertices of a DAG. So that if there is an edge from vertex i to vertex j then i appears before j in the linear ordering.

Ex:



The graph has no cycle i.e the graph is DAG ∴ the topological sort is possible.

(i) First find the DFS and push the visited vertices in the stack thus created a DFS travel stack.

- E_1
- C_2
- D_3
- A_4
- B_5

(2) Source Removal algorithm:

This is a direct implementation of decreases and conquer method.

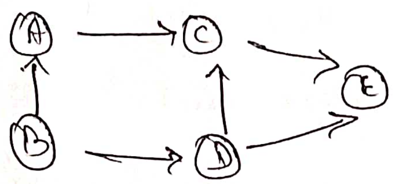
Following steps to be followed in this algorithm.

(i) from a given graph find a vertex with no incoming edges. Delete it along with all the edges outgoing from it. If there are more than one such vertices that break the tie randomly.

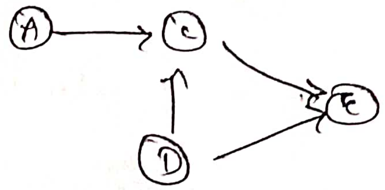
(2) Note the vertices that are deleted.

(3) All these recorded vertices give topologically sorted list.

Ex:

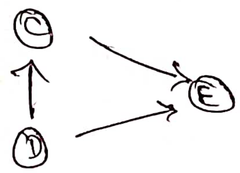


$\xrightarrow{\text{Delete } B}$



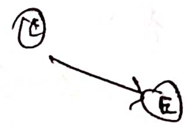
= B

$\xrightarrow{\text{Delete } A}$



= B, A

$\xrightarrow{\text{Delete } D}$



= B, A, D

$\xrightarrow{\text{Delete } C}$ E = B, A, D, ~~E~~ C

$\xrightarrow{\text{Delete } E}$ = B, A, D, C, E.

Hence the list after topological sorting will be B, A, D, C, E.

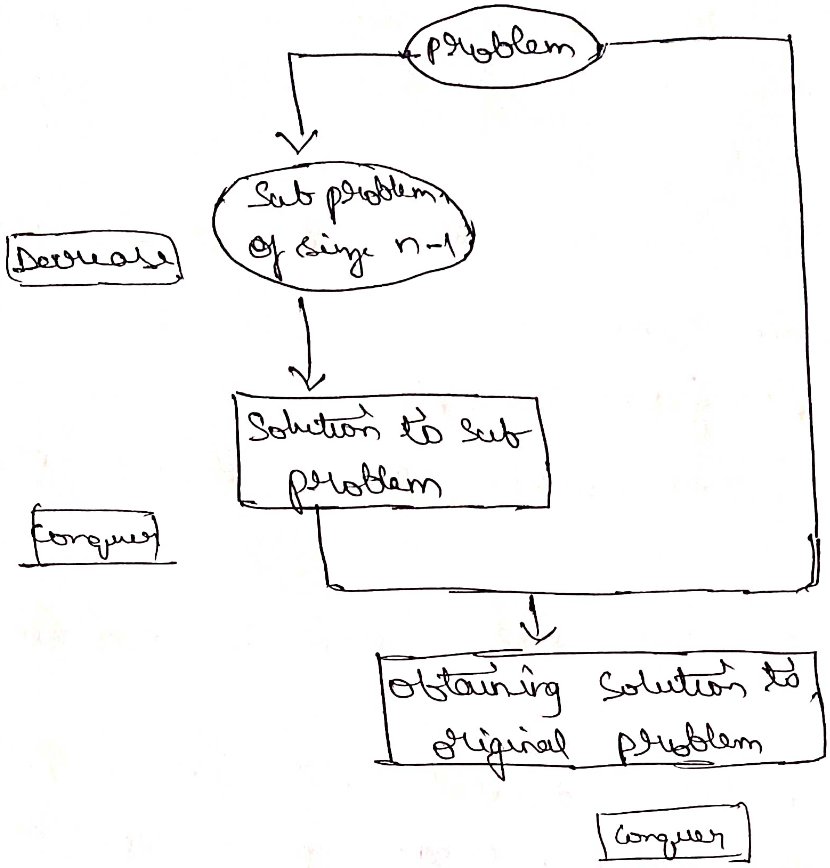
(2) what are the three major variations of decrease and conquer technique? Explain with an example for each.

→ There are 3 major variations of decrease and conquer.

(a) Decrease by constant (b) Decrease by a constant factor.

(c) variable size Decrease.

(a) Decrease by constant: In this method the size of the instance is reduced by same constant on each ~~iteration~~ iteration of the algorithm generally this constant is equal to one.

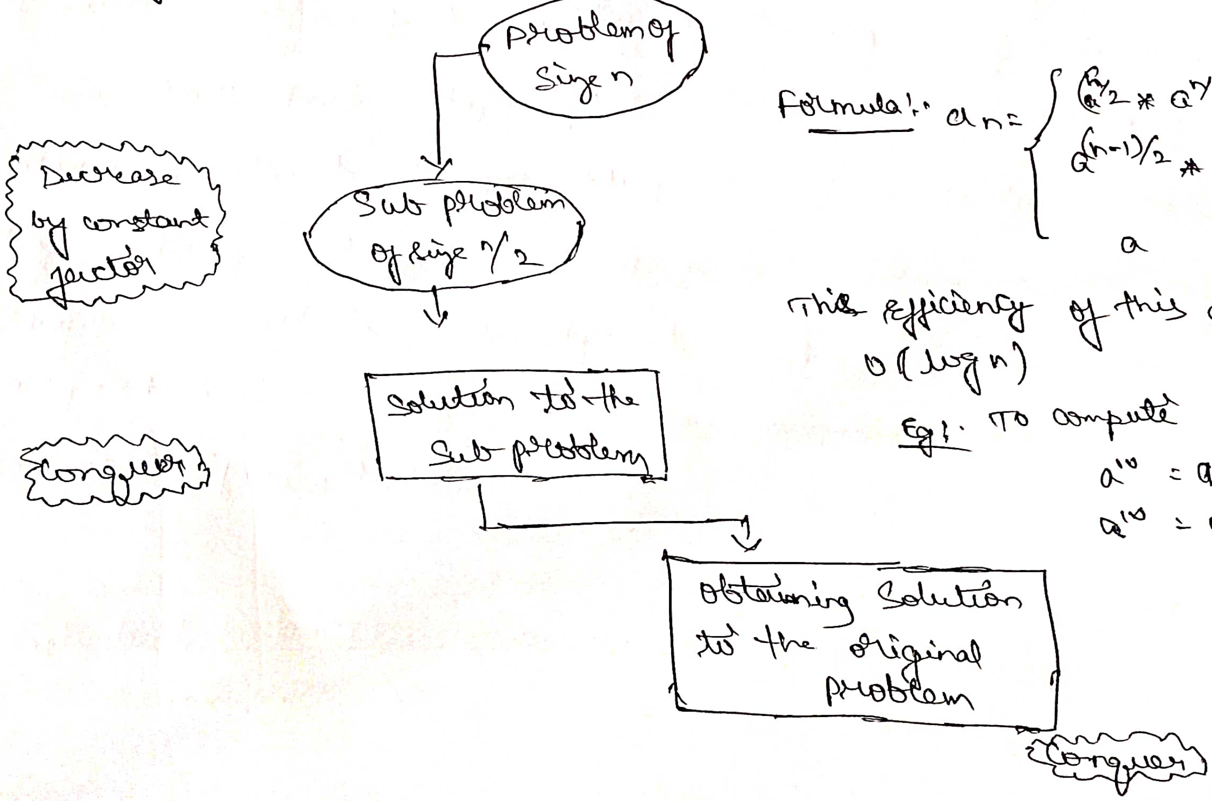


Formula: $a^n = a^{n-1} \cdot a$

Eg: $a^{10} = a^{10-1} \cdot a = a^9 \cdot a$

(b) Decrease by a constant factor:

Decrease by a constant factor decrease the instant size by half or by some other fraction.



Formula: $a^n = \begin{cases} (a^{n/2} * a^{n/2}) & \text{if } n \text{ is Even} \\ a^{(n-1)/2} * a^{(n-1)/2} * a & \text{if } n \text{ is odd} \\ a & \text{if } n = 1 \end{cases}$

This efficiency of this algorithm is $O(\log n)$

Eg: To compute a^{10} we can write

$a^{10} = a^{10/2} * a^{10/2}$

$a^{10} = a^5 * a^5$

(c) variable Size decrease :- In variable size decrease method the size reduction pattern varies from one iteration of an algorithm to another.

Eg :- Finding GCD of two numbers using Euclid's algorithm.

Formula for finding GCD is

$$gcd(m, n) = gcd(n, m \bmod n)$$

In finding GCD the two numbers go on varying until we get the GCD value.

③ what is job sequencing with deadline problem? Explain with an Example.

> we are given a set of n jobs. Associated with job i is an integer deadline $d_i \geq 0$ and a profit $P_i > 0$, for any job i the profit P_i is earned if the job is completed by its deadline. To complete a job one process the job on a machine for one ~~unit with of~~ unit of time. Only one machine is available for processing jobs. A feasible solution for this problem is a subset J of jobs such that each job in this subset can be completed by its deadline. The value of a feasible solution J is the sum of the profits of the jobs in J or $\Sigma_{i \in J} P_i$. An optional solution is a feasible solution with maximum value. Here again, since the problem involves the identification of a subset. It fits the subset paradigm.

Eg :- Let $n = 4$ $(P_1, P_2, P_3, P_4) = (100, 10, 15, 27)$ and $(d_1, d_2, d_3, d_4) = (2, 1, 2, 1)$ The feasible and their values are

	feasible solution	processing sequence	value
1	(1,2)	2,1	110
2	(1,3)	1,3 or 3,1	115
3	(1,4)	4,1	127
4	(2,3)	2,3	25
5	(3,4)	4,3	42
6	1	1	100
7	2	2	10
8	3	3	15
9	4	4	27

Solution 3 is optimal. In this solution only jobs 1 and 4 are processed and the value is 127. These jobs must be processed in the order job 4 followed by job 1. Thus processing of job 4 begins at the time zero and that of job 1 is completed at time 2.

(4) Solve the greedy knapsack problem where $m = 10$ & $n = 4$
$p(40, 42, 25, 12)$, $w = (4, 7, 5, 3)$

→ **Step 1:-** finding ration

$\frac{p_1}{w_1} = \frac{40}{4} = 10$ $\quad \frac{p_2}{w_2} = \frac{42}{7} = 6$ $\quad \frac{p_3}{w_3} = \frac{25}{5} = 5$ $\quad \frac{p_4}{w_4} = \frac{12}{3} = 4$

Step 2:- Arrange the p/w in descending order,

$(10, 6, 5, 4) = (obj\,1, obj\,2, obj\,3, obj\,4)$

Step 3:-

profit	40	42	25	12
weight	4	7	5	3
p/w	10	6	5	4
x_i	1	6/7	0	0

$$\Sigma x_i w_i = x_1 w_1 + x_2 w_2 + x_3 w_3 + x_4 w_4$$

$$= 1 \times 4 + \frac{6}{\cancel{7}} \times \cancel{7} + 0 + 0$$

$$\Sigma x_i w_i = 4 + 6 = 10$$

$$\therefore \Sigma x_i w_i \leq m.$$

profit can be calculated by

$$\Sigma x_i p_i = x_1 p_1 + x_2 p_2 + x_3 p_3 + x_4 p_4$$

$$= 1 \times 40 + \frac{6}{\cancel{7}} \times \cancel{42}^{7} + 0 + 0$$

$$\Sigma x_i p_i = 40 + 36 = 76$$

$\therefore$ Total profit $= 76$.

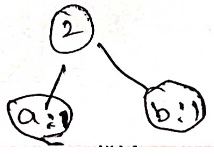
(5) Explain Huffman coding algorithm with an example Show the construction of huffman tree and generate the huffman code using this tree.

Step 1: Initialize n one-node trees and label them with the characters of the alphabet. Record the frequency of each character in its tree's root to indicate the tree's weight.

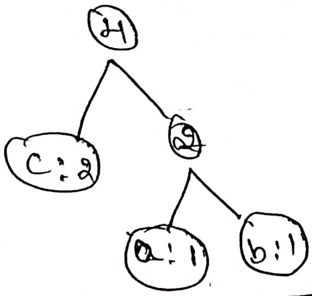
Step 2: Repeat the following operation until a single tree is obtained. find two trees with the smallest weight. make them left and right sub tree of a new tree and record the sum of their weights in the root of the new tree as its weight.

Ex:

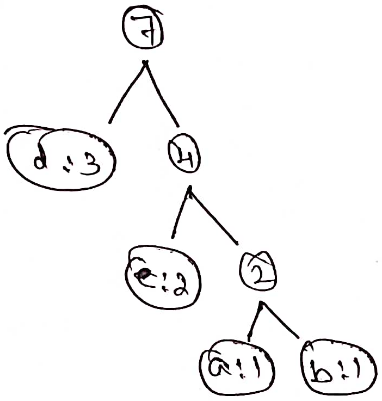
a:1	b:1	c:2	d:3	e:5	f:8	g:13	h:21



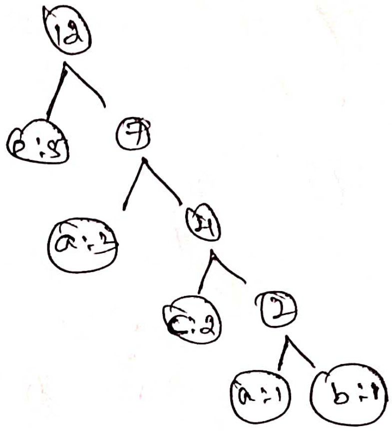
c:2	2	d:3	e:5	f:8	g:13	h:21



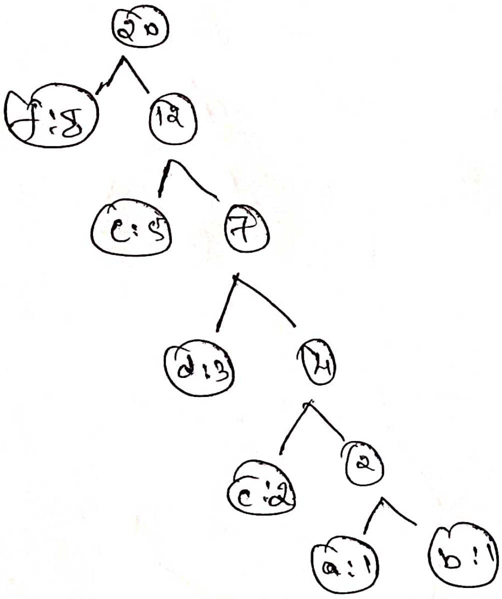
d:3	4	e:5	f:8	g:13	h:21



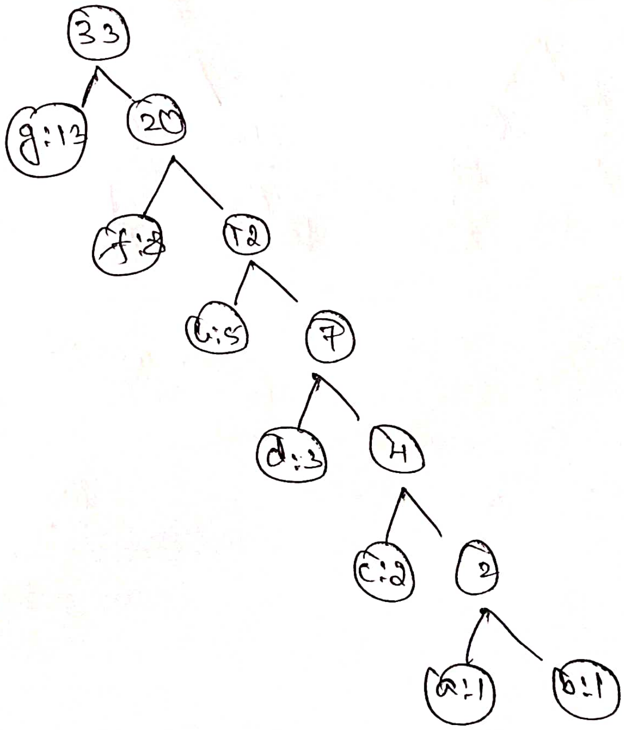
e:5	7	f:8	g:13	h:21

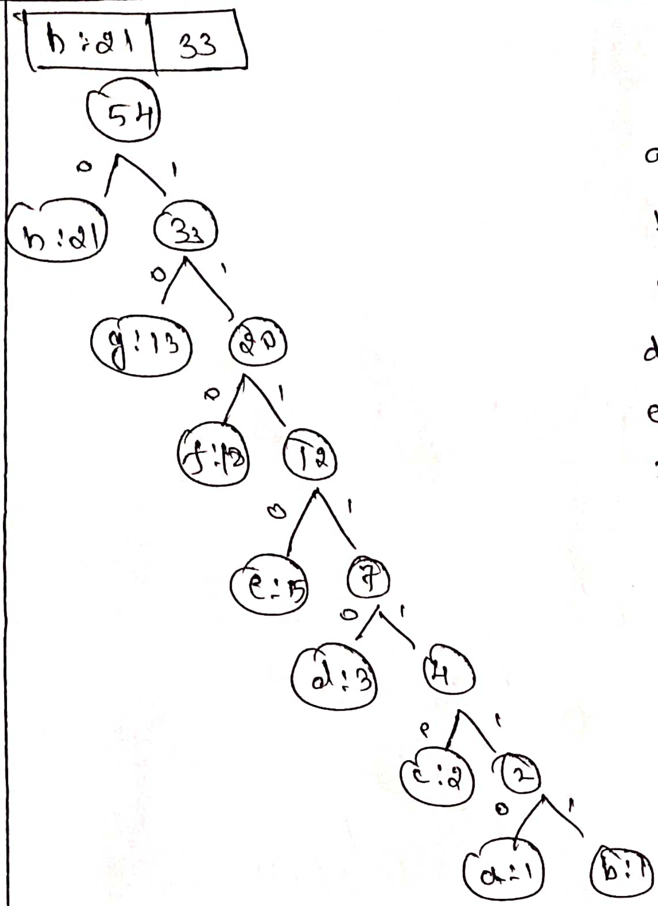


f:8	12	g:13	h:21



g:13	20	h:21





a:1 = 111110

b:1 = 111111

c:2 = 11110

d:3 = 1110

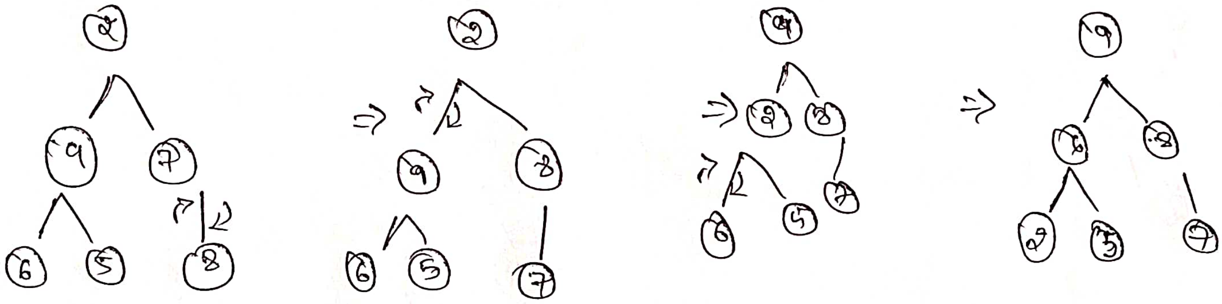
e:5 = 1110

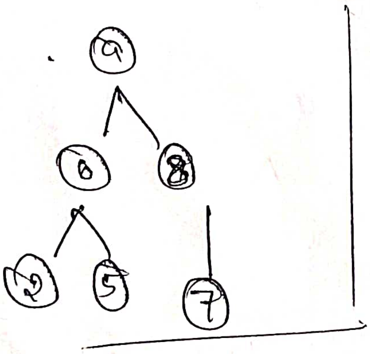
f:8 = 110

g:13 = 10

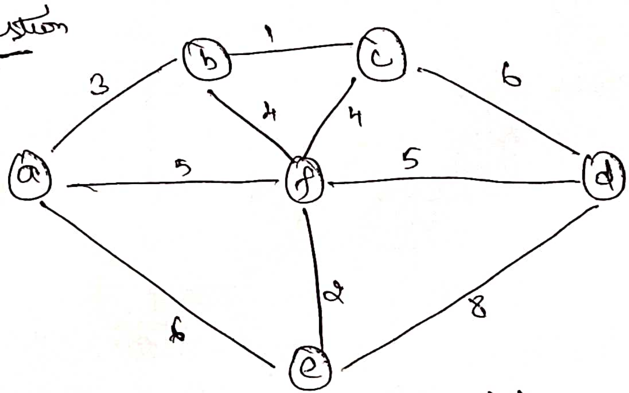
h:21 = 0

(6) Sort the array 2, 9, 7, 6, 5, 8 by ~~heaps~~ heap sort.





(7) 7th question



Apply prims and kruskals algorithm to find the MST of the graph given below.

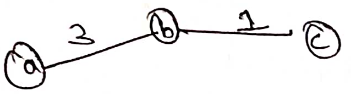
prims algorithm: * Select an Edge starting from vertex a with

7

minimum path length.

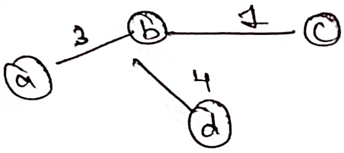
Total path = 3

* Select next adjacent Edge of minimum path length.



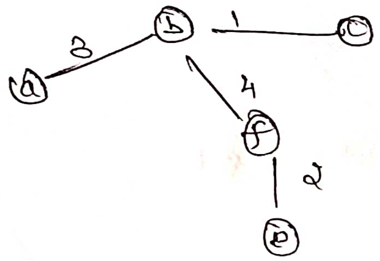
path length = 4

* Select next adjacent Edge of minimum path length.



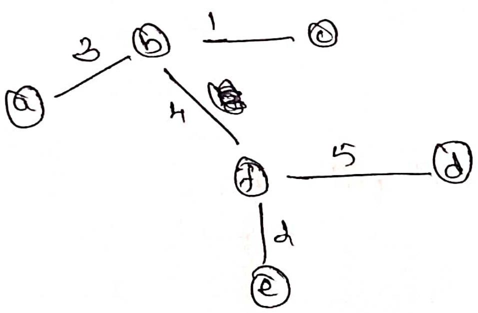
total path = 8

* Select next adjacent edge of minimum path length.



total path = 10

* Select next adjacent Edge of minimum path length.

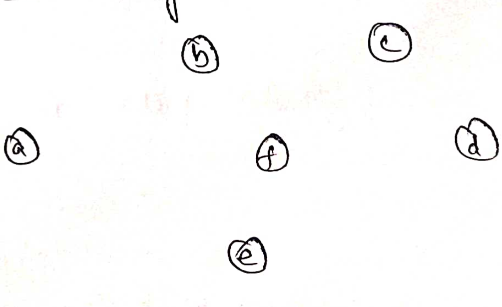


total path length = 15

This is required MST.

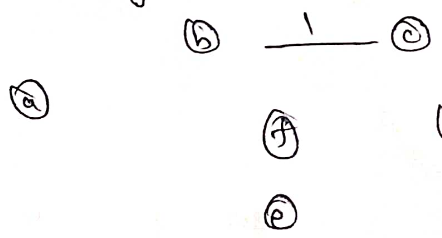
=> Kruskal's Algorithm.

Step 1:



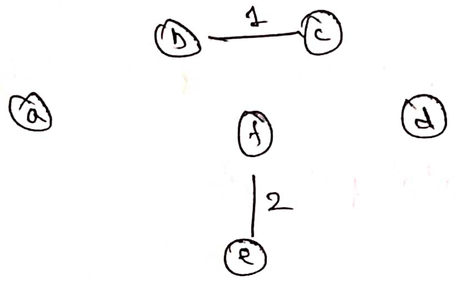
* Select an Edge with minimum weight

Step 2:-



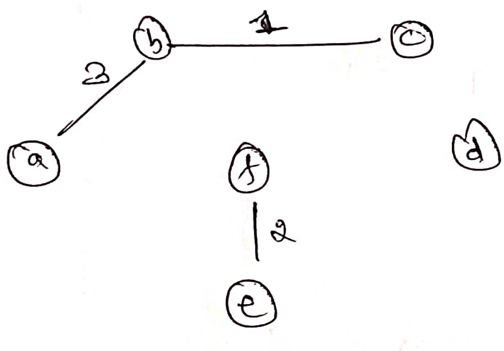
total path =1

* then we select the next minimum weighted edge. It is not necessary that selected edge is adjacent



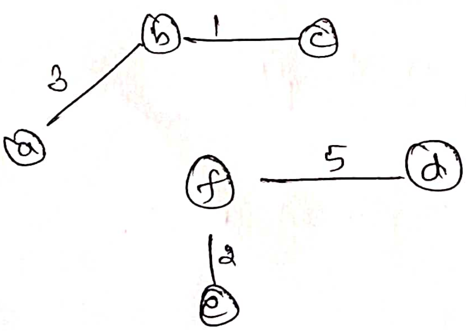
total path =3

*



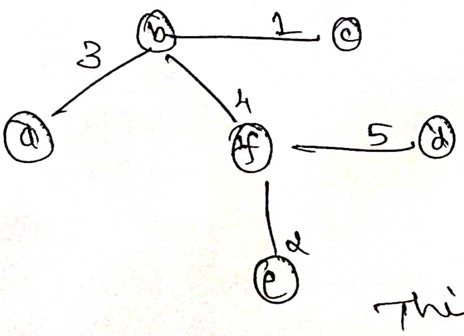
total path = 6

*



total path =10

*



Total path = 15

This is required MST //.

8. Find Solution generated by job sequencing problem with deadlines for 7 jobs given profit 3, 5, 20, 18, 1, 6, 30 & deadlines 1, 3, 4, 3, 2, 1, 2 respectively.

$(P_1, P_2, P_3, P_4, P_5, P_6, P_7)$
$(3, 5, 20, 18, 1, 6, 30)$

$d_1, d_2, d_3, d_4, d_5, d_6, d_7$
$1, 3, 4, 3, 2, 1, 2$

Solution: 8 is optimal. In this solution only jobs 3 and 4 are processed and value is 38 (Acc to table which is Solved).

	feasible Solution	processing sequence	value
1	(1, 2)	1, 3	8
2	(1, 3)	1, 4	23
3	(1, 4)	1, 3	21
4	(1, 5)	1, 2	4
5	(1, 6)	1, 1	9
6	(1, 7)	1, 2	33
7	(2, 3)	3, 4	25
8	(3, 4)	1, 3	38
9	(4, 5)	3, 2	19
10	(5, 6)	2, 1	7
11	(6, 7)	1, 2	36
12	(1)	1	3
13	(2)	3	5
14	(3)	4	20
15	(4)	3	18
16	(5)	2	1
17	(6)	1	6
18	(7)	2	30