

Unit : 4 : Greedy Algorithm

* Explain Greedy Algorithm with its characteristics.

=> Greedy Algorithm is used to select things which is best at that moment.

Greedy Algorithm some time not gives optimal solution.

Greedy Algorithm is a technique to select the thing which is best at a moment.

This method is basically used to determine the feasible solution that may or may not be optimal.

Greedy Algorithm can take decision on the basis of the currently available information.

Some time Greedy algorithm does not worry about the future decision.

Greedy Algorithm is used to find the shortest path in graph.

→ Characteristics of Greedy Algorithm:

- 1 Simple Algorithm
- 2 Faster to Execute
- 3 Efficiency
- 4 Less computing resource.

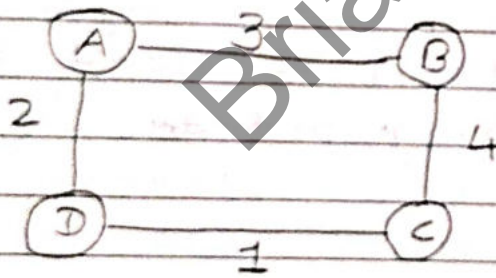
* Explain Minimum Spanning Tree with example.

=> A Spanning Tree of a connected graph G contains all the nodes and has the edges which connects all the nodes.

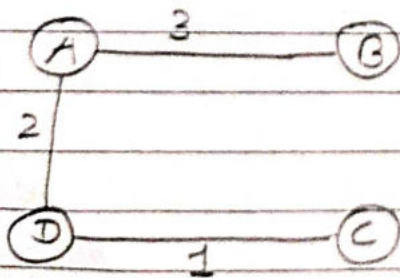
In Spanning Tree, there are no cycle in the graph.

A Minimum Spanning tree is found when the edges are picked to minimize the total cost.

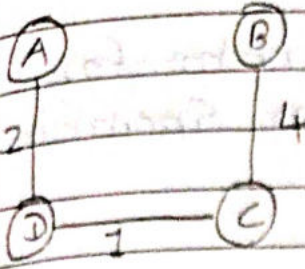
Ex.



Possible Spanning Tree:

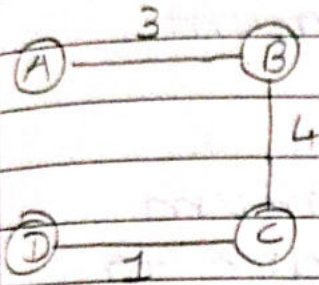


$$\text{Cost} = 3 + 2 + 1 = 6$$



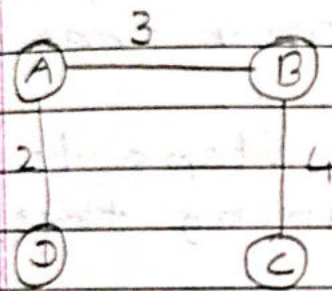
$$\text{Cost} = 4 + 2 + 1$$

$$= 7$$



$$\text{Cost} = 4 + 3 + 1$$

$$= 8$$



$$\text{Cost} = 4 + 3 + 2$$

$$= 9$$

Among the Four Possible Tree,
We get Minimum Spanning
Tree with cost = 6.

* Explain Prim's Algorithm for Finding the Minimum Spanning Tree.

=> Using Prim's Algorithm, we can find the minimum Spanning Tree.

For Finding the Minimum Spanning Tree, we have to Remove All the Parallel edges and Self loops of any tree.

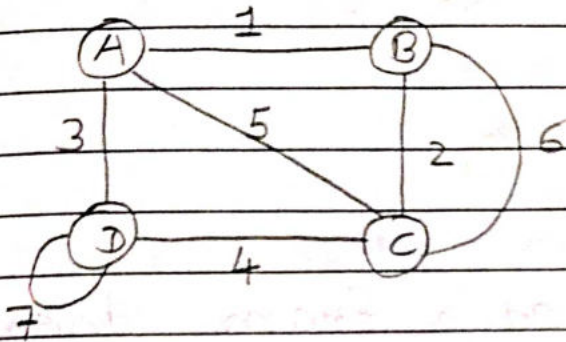
This are the Basic Steps to Find Minimum Spanning tree.

Steps:

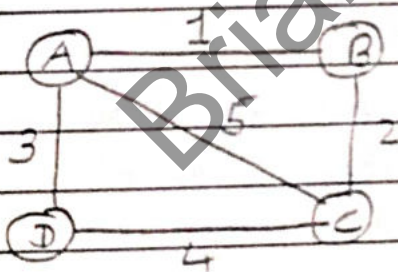
- 1 Create Edges list of the Given Graph.
- 2 Create skeleton for Spanning Tree.
- 3 Select Minimum edges from the edges list.
- 4 After that select one end point from the minimum edges.

5 Repeat the steps until $n-1$ edges are added.

Ex.



For Find the Minimum Spanning tree, we have to remove self loop and Parallel edges from the graph.



Create edges list From the Graph.

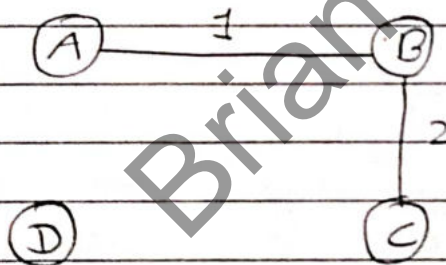
Edge	Weight
AB	1
BC	2
CD	4
DA	3
AC	5

Create Skeleton For Graph and Select the minimum weight edges.



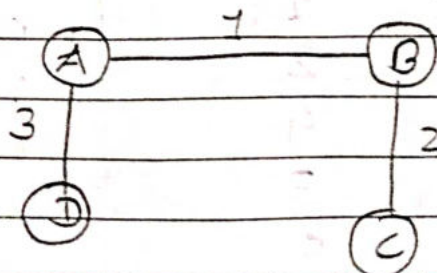
Select the edge from edges list with A and B endpoint with minimum weight.

Here, BC (2) and DA (3).
So, take BC edges.



Select the edge from edges list with A, D and C endpoint.

Here, DA (3), DC (4).
So, take DA edges.



Here, We get Minimum Spanning Tree with cost = 6

* Explain Kruskal's Algorithm for Finding the Minimum Spanning Tree.

⇒ Using Kruskal's Algorithm, we can find the minimum Spanning Tree.

For Finding the minimum Spanning Tree, we have to Remove All the Parallel edges and self loops of any trees.

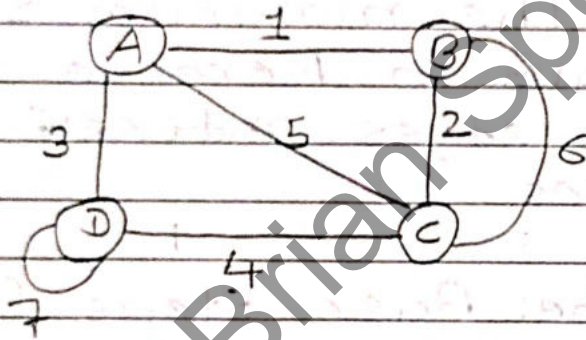
This are the Basic Steps for Finding the Minimum Spanning tree.

Steps:

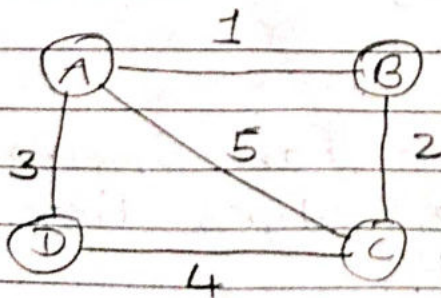
- 1 Create the edges list of the Given Graph.
- 2 Create skeleton for Spanning tree and sort the edges list according to its weight.

- 3 Take the edge at the top of the edge list and Remove from the table.
- 4 Select the second edge from the edge list but it is not create cycle.
- 5 Repeat the steps until $n-1$ edges are added.

Ex.



For Finding the Minimum Spanning tree, we have to remove self loop and Parallel edges from the graph.



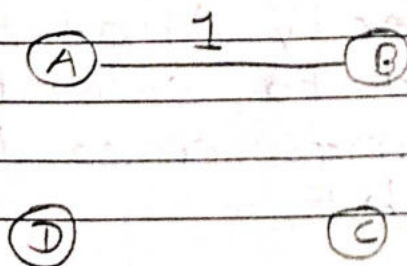
Create edges list from the Graph

Edge	Weight
AB	1
BC	2
CD	4
DA	3
AC	5

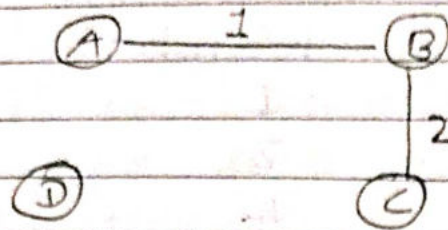
Sort the edges list, according to its weight.

Edge	Weight
AB	1
BC	2
DA	3
BC	4
AC	5

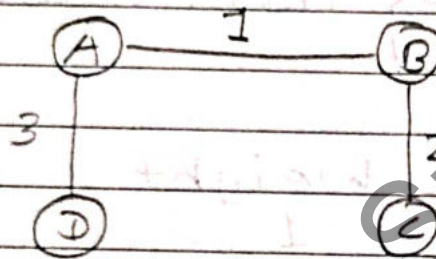
Create a skeleton for tree and take First edge and Remove From the list.



Select Second edge from the edges list.



Select third edge from the edges list.



Here, we get Minimum Spanning tree with cost = 6.

* Explain Fractional Knapsack Problem with example.

=> Fractional Knapsack Problem is follow Greedy Approach which is used to select n different things according to its weight.

This are the basic step to Perform Fractional knapsack Problem.

Steps: "

1 Calculate the Density of every item.

2 Sort the item according to its density.

3 Take Total Weight and Total Profit = 0.

check for every item

$$\text{Weight (Item)} + \text{Total Weight} \leq W$$

than Select this item.

4 else Follow this step,

$$x[i] = \frac{W - \text{item Weight}}{\text{Weight}}$$

$$\text{Profit} = \text{Profit} + (x[i] / w[i]) * v[i]$$

i = item number.

Ex. Total Weight $W = 16$ Kg

Item	Weight (w)	Value
1	6	6
2	10	2
3	3	1
4	5	8
5	1	3

\Rightarrow Here, Given Total Weight $W = 16$ kg which means we have to select only 16 kg Item.

According to Fractional Knapsack Problem,

- Step - 1 : Find the value of each Item or density.

Item	w	v	Density
1	6	6	1
2	10	2	0.20
3	3	1	0.33
4	5	8	1.60
5	1	3	3

- Step - 2 : Sort the Item, according to its Density.

Item	W	V	Density
5	1	3	3
4	5	8	1.60
1	6	6	1
3	3	1	0.333
2	10	2	0.20

- Step - 3: Pick the First Item in the table and check

$$\begin{matrix} \text{Item} + \text{Total} & \leq & \text{Weight} \\ \text{Weight} & & \text{Weight} \\ 1 + 0 & \leq & 16 \end{matrix}$$

Condition is follow so, take this item.

Item	W	V	Total Weight	Total Profit
-	-	-	0	0
5	1	3	1	3

- Step: 4: Take Second Item with Total Weight - 1

$\therefore 5 + 1 \leq 16$, Condition is follow

Item	W	V	Total Weight	Total Profit
-	-	-	0	0
5	1	3	1	3
4	5	8	6	11

- Step 5: Take Third Item with weight 6

$\therefore 6 + 6 \leq 16$, \therefore Condition is follow.

Item	W	V	Total Weight	Total Profit
-	-	-	0	0
5	1	3	1	3
4	5	8	6	11
1	6	6	12	17

- Step 6: take Fourth Item with weight 12

$\therefore 3 + 12 \leq 16$, \therefore Condition is follow.

Item	W	V	Total Weight	Total Profit
-	-	-	0	0
5	1	3	1	3
4	5	8	6	11
1	6	6	12	17
3	3	1	15	18

- Step 7: take Last item with weight 15

$\therefore 15 + 10 \not\leq 16$, \therefore Condition is not follow.

$$\text{Find, } X[2] = 16 - 15 = 1$$

$$\text{Profit} = 18 + (0.20 * 2) \\ = 18.4$$

Item	W	V	Total Weight	Total Profit
-	-	-	0	0
5	1	3	1	3
4	5	8	6	11
1	6	6	12	17
3	3	1	15	18
2	10	2	16	18.4

* Explain Coin Change Problem with example.

=> Coin Change Problem is used Greedy approach which is used to give certain amount of change.

In Coin Change Problem, we have to always select maximum value of coin.

Algorithm For Coin Change Problem :

1 Sort the coin value in descending Order.

2 IF Current - Value \leq Remaining Value
Coin

than
total = total + Current - Value
Coin Coin Coin

3 else Increment

4 Repeat the step until
Remaining value = 0

Ex. Available coin 20, 10, 5, 1
and we have to make ₹3
Value.

=> Sorted coin value in descending
Order \rightarrow 20, 10, 5, 1

take 20 \leq Make value

20 \leq ₹3. So take 20

and Remaining
value ₹3 - 20 = 23

~~total coin = 20~~

take 20 $<$ Remaining value

$\therefore 20 \leq 23$, So take 20 coin
 Remaining value = $23 - 20$
 $= 3$

take $20 \leq 3$, Condition is False
 So, Increment value.

take, $10 \leq 3$, Again condition is False, So, Increment value.

take, $5 \leq 3$, Again condition is False, So, Increment value.

take, $1 \leq 3$, Condition is true
 So take 1 coin and
 Remaining value = $3 - 1 = 2$

Again take, $1 \leq 2$, So take 1
 coin and Remaining
 value = $2 - 1 = 1$

Again take, $1 \leq 1$, So take 1
 coin and Remaining value = $1 - 1 = 0$

So, Total
 Coin = $[20, 20, 1, 1, 1]$

Total Coin = $1 + 1 + 1 + 1 + 1$
 $= 5$ Coin