

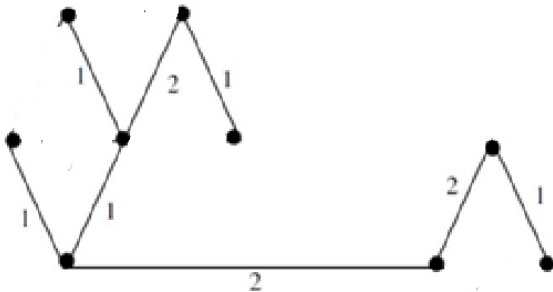
Course/ Batch: BTech/ SCSET
Course Code: CSET244
Semester: Even
Session: 2024-2025

Course Type: Core
Course Name: Design and Analysis of Algorithms

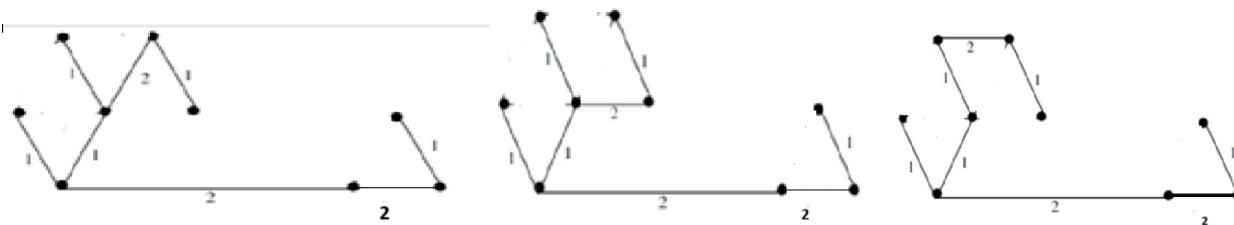
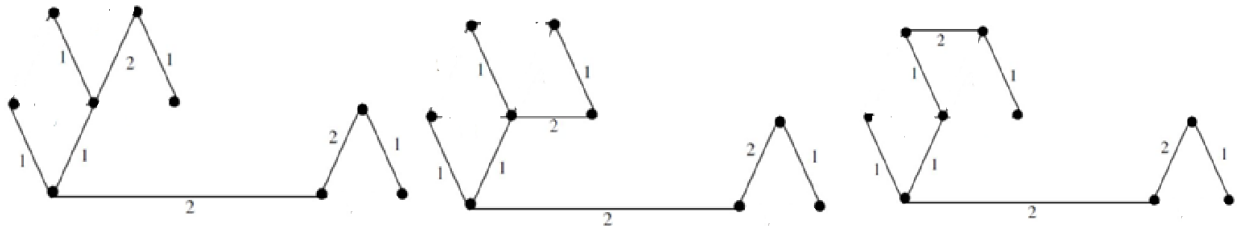
Tutorial Assignment: Week 7

Tutorial title: Minimum Spanning Tree

Q1. (a) 11



(b) 6 MST's can be possible.



Q2. You may discuss any algorithm of your choice. One relevant algorithm is discussed here.

Use disjoint set data structure (Union-Find operation) where you build a spanning tree and check if adding any edge creates a cycle by connecting two already connected components; if a new edge connects two nodes that are already in the same set, then a cycle exists.

```

Detect_Cycle(G)
{
  for each edge (u, v)
    check if the sets containing u and v are different
    if they are different
      merge the sets of u and v using the Union-Find operation
    if they are the same
      print('cycle exists')
}

```

Consider a graph with nodes A, B, C, D, E and edges (A, B), (B, C), (C, D), (D, E), (A, C).

Initially, each node is in its own set: {A}, {B}, {C}, {D}, {E}.

When processing edge (A, B), merge sets {A} and {B} to create {A, B}.

Processing (B, C) merges {A, B} and {C} to form {A, B, C}.

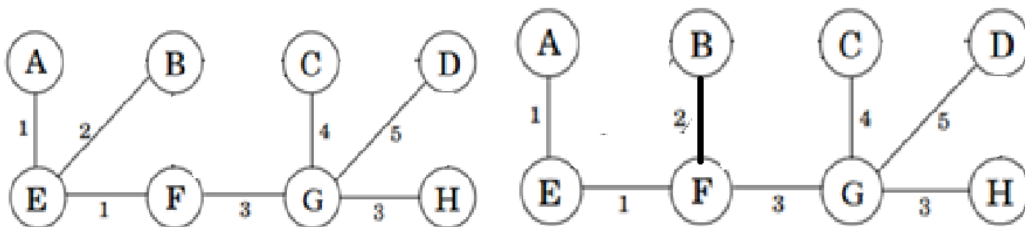
When processing (A, C), since A and C are already in the same set {A, B, C}, a cycle is detected.

Time Complexity

The time complexity of this approach is $O(E \log V)$ where E is the number of edges and V is the number of vertices, due to the Union-Find operations.

Q3. (a) 19

(b) More than one MST is possible. Two such MSTs are shown below:



(c) **Kruskal's Algorithm:**

The edges and weights of the graph is as follows:

Edge	AB	AE	BC	BE	BF	CD	CF	CG	DG	DH	EF	FG	GH
Weight	6	1	5	2	2	6	5	4	5	7	1	3	3

Step1: Arrange the weights according to the increasing order of their weights.

Edge	AE	EF	BE	BF	FG	GH	CG	BC	CF	DG	AB	ED	TDH
Weight	1	1	2	2	3	3	4	5	5	5	6	6	7

Step2: Select the edges with minimum weight which does not form a cycle.

Edge	AE	EF
Weight	1	1

Step3: Repeat the above step until all the vertices covered and number of edges = |Vertices|-1.

Added edges are as follows:

(i) AE, EF, BE, FG, GH, CG, DG

(ii) AE, EF, BF, FG, GH, CG, DG

Edge	AE	EF	BE	FG	GH	CG	DG
Weight	1	1	2	3	3	4	5

OR

Edge	AE	EF	BF	FG	GH	CG	DG
Weight	1	1	2	3	3	4	5

(d) One can start with any random node/vertex and add the corresponding edge with its weight.

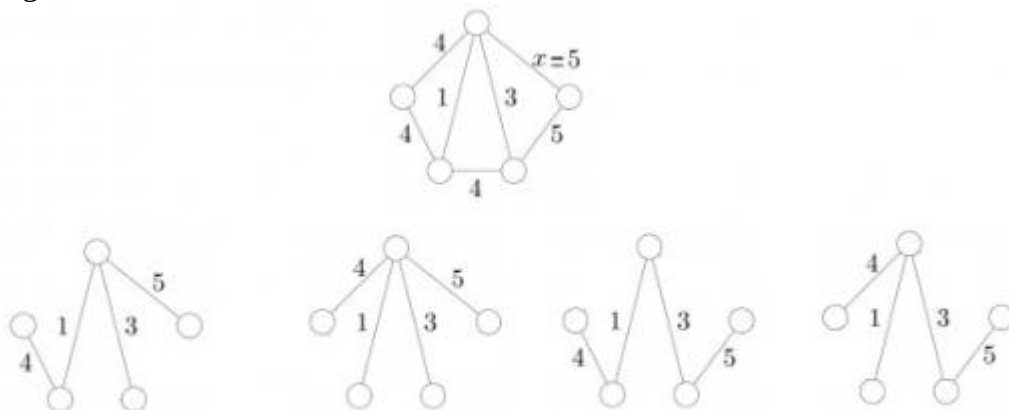
Q4. Consider the cycle ABC. AC and BC are part of minimum spanning tree. So, AB should be greater than $\max(AC, BC)$ (greater and not equal as edge weights are given to be distinct), as otherwise we could add AB to the minimum spanning tree and removed the greater of AC, BC and we could have got another minimum spanning tree. So, $AB > 9$.

Similarly, for the cycle DEF, $ED > 6$.

And for the cycle BCDE, $CD > 15$.

So, minimum possible sum of these will be $10 + 7 + 16 = 33$. Adding the weight of spanning tree, we get the total sum of edge weights = $33 + 36 = 69$

Q5. When $x = 5$, we get maximum 4 MWSTs



Value of $x = 5$

$\therefore$ 4 MWSTs possible.