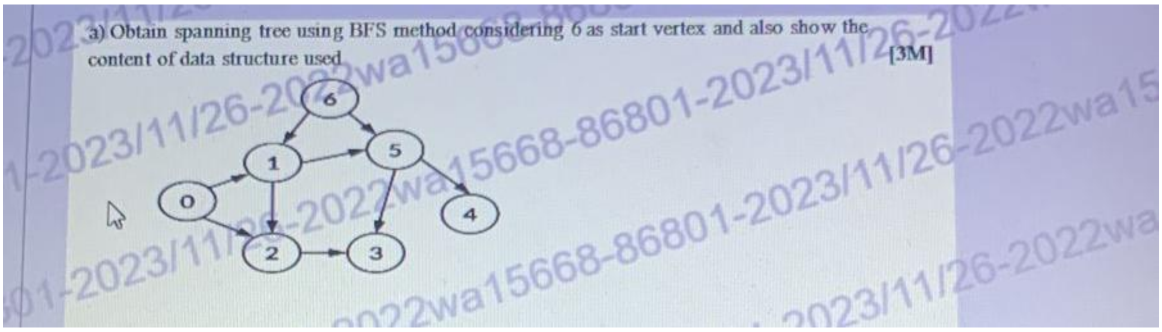


a) Obtain spanning tree using BFS method considering 6 as start vertex and also show the content of data structure used [3M]

Solution:

The spanning tree using the BFS (Breadth-First Search) method from node 6 as the start would include the following edges: 6-1, 6-5, 1-0, 1-2, 5-3, 5-4. The content of the data structure used, such as the queue for BFS, would include the nodes visited in sequence.

Content of the data structure (BFS queue) during the process:

// Start at node 6:

Queue: [6]

// Explore neighbors of node 6 (1 and 5)

Queue: [1, 5]

// Explore neighbors of node 1 (0 and 2):

Queue: [5, 0, 2]

// Explore neighbors of node 5 (3 and 4):

Queue: [0, 2, 3]

// Explore neighbors of node 0 (already visited):

Queue: [2, 3, 4]

// Explore neighbors of node 2 (3 is unvisited):

Queue: [3, 4]

// Explore neighbors of node 3 (4 is unvisited):

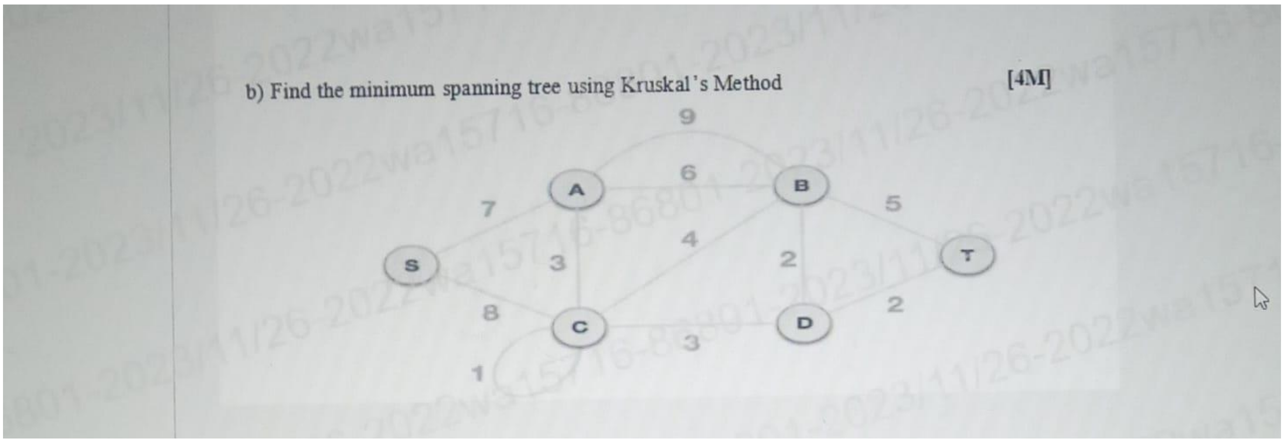
Queue: [4]

// Explore neighbors of node 4 (done):

Queue: []

The spanning tree edges are:

```
6-1, 6-5, 1-0, 1-2, 5-3, 5-4
```



b) Find the minimum spanning tree using Kruskal's Method [4M]

Solution:

In Kruskal's algorithm for finding the minimum spanning tree, we begin by sorting the edges based on their weights. The idea is to start with the smallest weighted edge and progressively add edges while ensuring that no cycles are formed. Let's break down the process for the given connections:

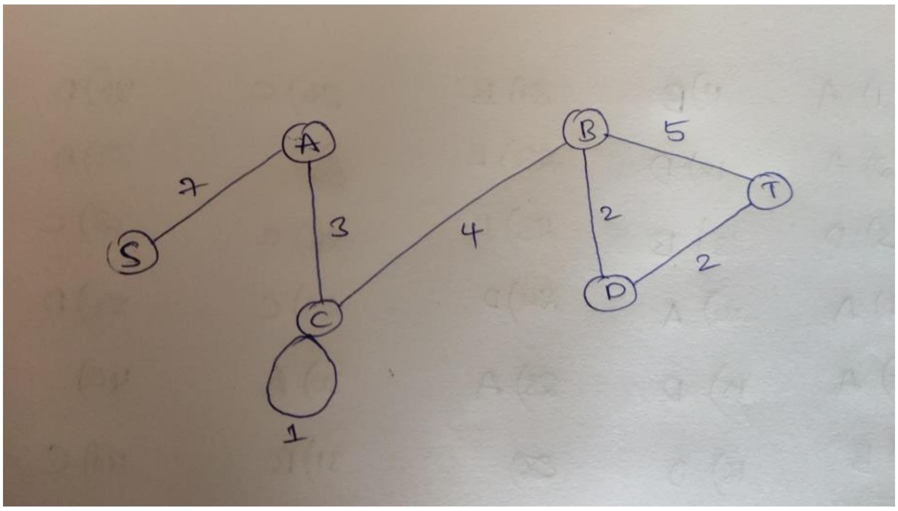
First, we sort these connections in ascending order based on their weights:

1. C-C: 1
2. B-D: 2
3. D-T: 2
4. C-A: 3
5. C-D: 3
6. C-B: 4
7. S-A: 7
8. A-B: 6, 9 (sorting both values)
9. S-C: 8
10. B-T: 5

Starting with the smallest weighted edge, we iteratively select edges without creating any cycles:

1. **C-C: 1** - Selecting the connection within C itself, a self-loop.
2. **B-D: 2** - Adding the edge between B and D.
3. **D-T: 2** - Including the connection between D and T.
4. **C-A: 3** - Adding the link between C and A.
5. **C-D: 3** - Including the edge from C to D.
6. **C-B: 4** - Selecting the link between C and B.
7. **S-A: 7** - Connecting S and A.
8. **B-T: 5** - Adding the edge from B to T.

The minimum spanning tree diagram is:



The selected edges form the minimum spanning tree, ensuring connectivity between all nodes without creating any cycles. This configuration has the minimum total weight among the possible spanning trees for this graph.