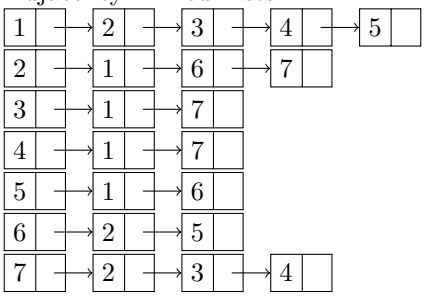
Traversing Solution

hxxz46

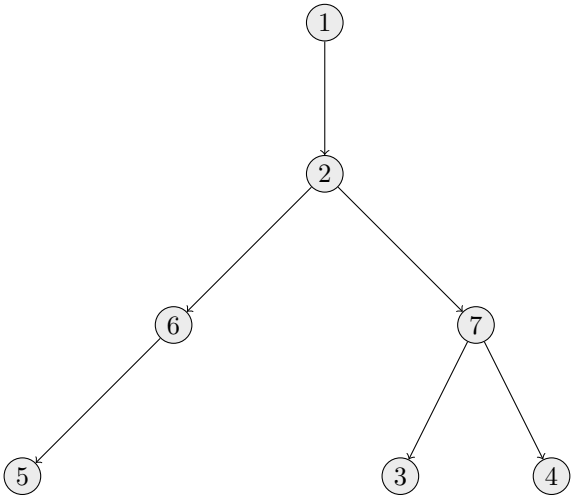
B.2.1: Adjacency Matrix:

$$\begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} \end{array}$$

Adjacency Linked Lists:



B.2.2:
DFS Tree:



The order in which the vertices are reached for the first time is:

$$1, 2, 6, 5, 7, 3, 4$$

The order in which the vertices become dead ends is:

$$5, 6, 3, 4, 7, 2, 1$$

1. We start at vertex 1 it is pushed onto the traversal stack [1].
2. From 1, it is possible to visit 2,3,4 or 5 next. Breaking ties numerically, we visit 2 and push it to the traversal stack [2,1]. We have not fully finished exploring 1 so it remains on the stack.
3. From 1, it is possible to visit 6 or 7 next. Breaking ties numerically, we visit 6 and push it to the traversal stack [6,2,1]. We have not fully finished exploring 2 so it remains on the stack.
4. From 6, it is only possible to visit 5. We push it to the traversal stack [5,6,2,1]. We have not fully finished exploring 6 so it remains on the stack.
5. We have reached a dead end at 5, there are no new vertices to explore. We pop it from the traversal stack [6,2,1] and consider 6.
6. We have also reached a dead end at 6, there are no new vertices to explore. We pop it from the traversal stack [2,1] and consider 2.
7. From 2, it is possible to reach 7 which has not yet been visited. We push it to the traversal stack [7,2,1]. We have not fully finished exploring 2 so it remains on the stack.
8. From 7, it is possible to visit 3 or 4. Breaking ties numerically, we visit 3 and push it to the traversal stack [3,7,2,1]. We have not fully finished exploring 7 so it remains on the stack.

9. We have reached a dead end at 3, there are no new vertices to explore. We pop it from the traversal stack [7,2,1] and consider 7.

10. From 7, it is possible to visit 4. We push it to the traversal stack [4,7,2,1]. We have not fully finished exploring 7 so it remains on the stack.

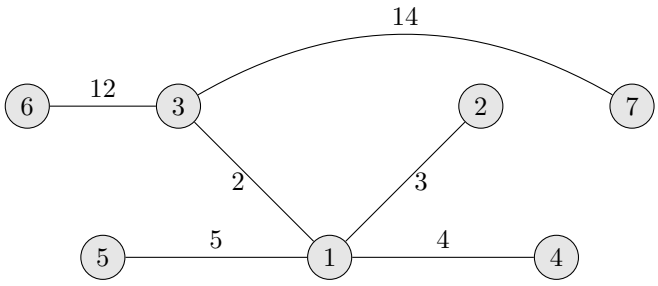
11. We have reached a dead end at 4 as there are no new vertices to explore. We pop it from the traversal stack [7,2,1] and consider 7.

12. We have reached a dead end at 7 as there are no new vertices to explore. We pop it from the traversal stack [2,1] and consider 2.

13. We have reached a dead end at 2 as there are no new vertices to explore. We pop it from the traversal stack [1] and consider 1.

14. We have reached a dead end at1 as there are no new vertices to explore. We pop it from the traversal stack [] and finish as there are no nodes remaining in the stack.

B.2.3:



The total weight of this tree is 35.

1. Greedily choose the edge with minimum weight. In this case it is the edge from 1 to 2 with weight 2.

2. Greedily choose the edge with the next smallest weight. In this case it is the edge from 1 to 3 with weight 3. Adding this edge does not create a cycle so we add it to the minimum spanning tree.

3. Greedily choose the edge with the next smallest weight. In this case it is the edge from 1 to 4 with weight 4. Adding this edge does not create a cycle so we add it to the tree.

4. Choose the edge with the next smallest weight. In this case it is the edge from 1 to 5 with weight 5. Adding this edge does not create a cycle so we add it to the tree.

5. Choose the edge with the next smallest weight. In this case it is the edge from 2 to 6 with weight 12. Adding this edge does not create a cycle so we add it to the tree.

6. Choose the edge with the next smallest weight. In this case it is the edge from 2 to 7 with weight 14. Adding this edge does not create a cycle so we add it to the tree.

7. At this point, add any further edges would create a cycle. The minimum spanning tree is a connected graph. Therefore, we finish as we have already produced the minimum spanning tree.