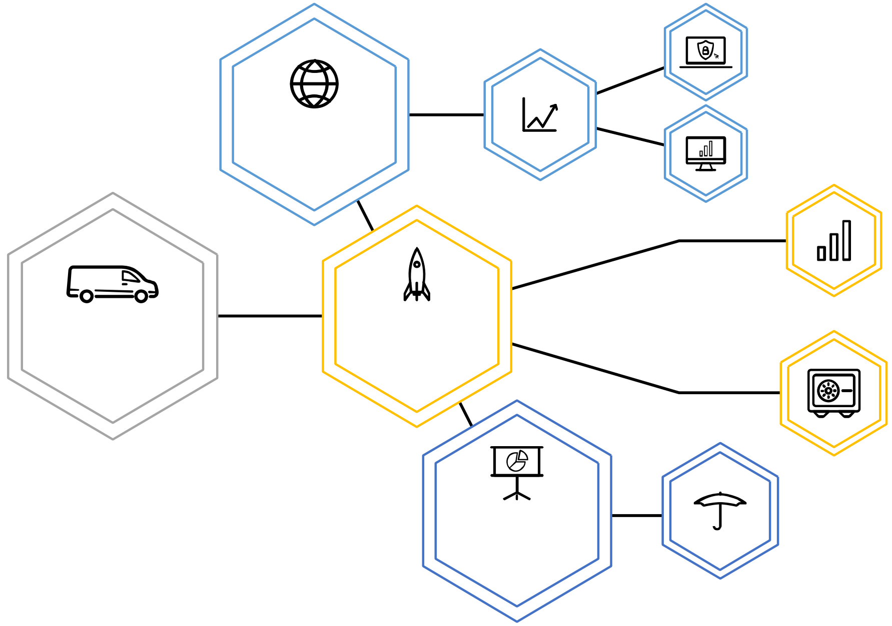


AV121: Data Structures and Algorithms

Tutorial-04

Tutorial 04 – Plan

- 2-3 Trees
- Graph ADT
  - → Concepts
  - → BFS
  - → DFS
  - → Spanning Trees

2-3 Trees

- All leaves of a 2-3 Tree have to be at the same level

    A. True

    B. False

2-3 Trees

- 2-3 Trees are binary search trees
  - A. True
  - B. False

2-3 Trees

- Which of the following statements are true

A. 2–3 tree is a tree data structure

B. Every internal node (Node with children) meets either of the following:
   - → Has two children (2-node) and one data element
   - → Has three children (3-nodes) and two data elements

C. All leaves are at the same level

D. All data is stored in sorted order

E. All of the above

2-3 Trees

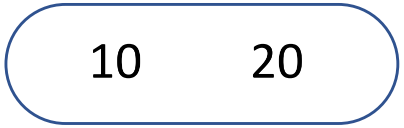
- Height of a 2-3 tree is bounded by
  A. O(log n)
  B. O(n)
  C. O(nlog n)
  D. None of the above

2-3 Trees

- Construct a 2-3 tree by inserting the following elements in the given order
  - → 10, 20, 30, 40, 50, 60, 70

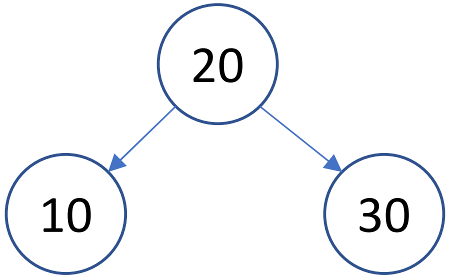
2-3 Trees

- Construct a 2-3 tree by inserting the following elements in the given order
  → 10, 20, 30, 40, 50, 60, 70



2-3 Trees

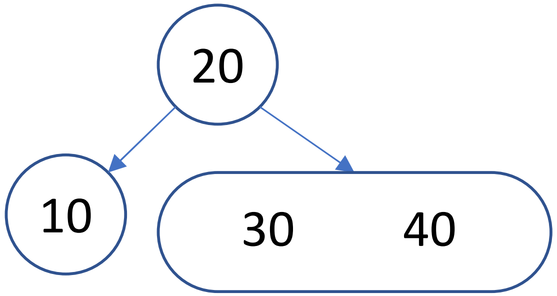
- Construct a 2-3 tree by inserting the following elements in the given order
  - → 10, 20, 30, 40, 50, 60, 70



2-3 Trees

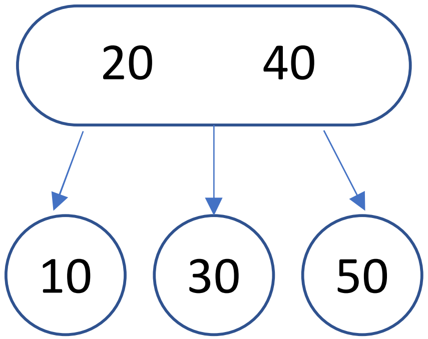
- Construct a 2-3 tree by inserting the following elements in the given order

  → 10, 20, 30, 40, 50, 60, 70



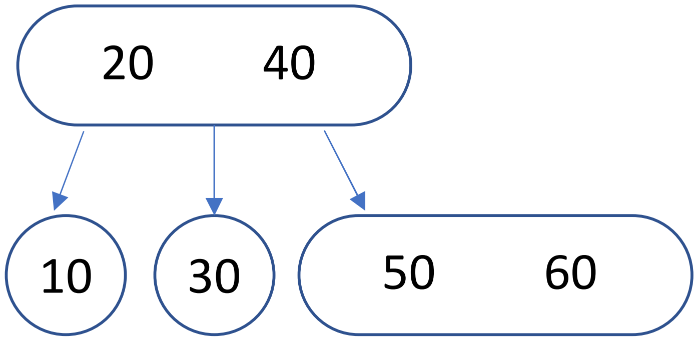
2-3 Trees

- Construct a 2-3 tree by inserting the following elements in the given order
  - → 10, 20, 30, 40, 50, 60, 70



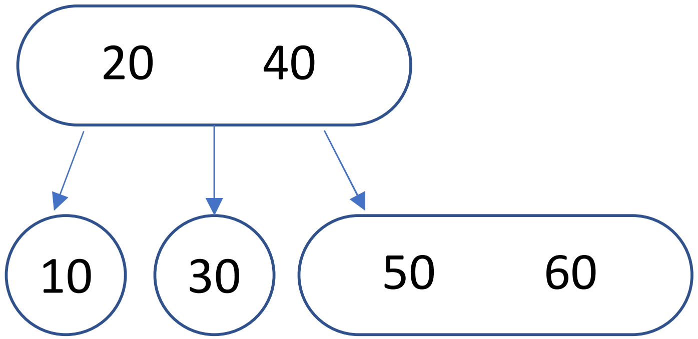
2-3 Trees

- Construct a 2-3 tree by inserting the following elements in the given order
  - → 10, 20, 30, 40, 50, 60, 70



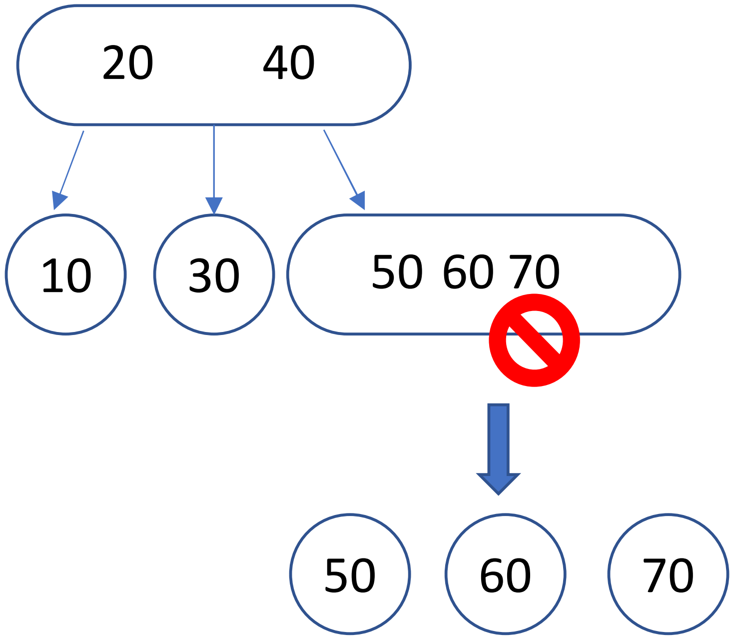
2-3 Trees

- Construct a 2-3 tree by inserting the following elements in the given order
  - → 10, 20, 30, 40, 50, 60, 70



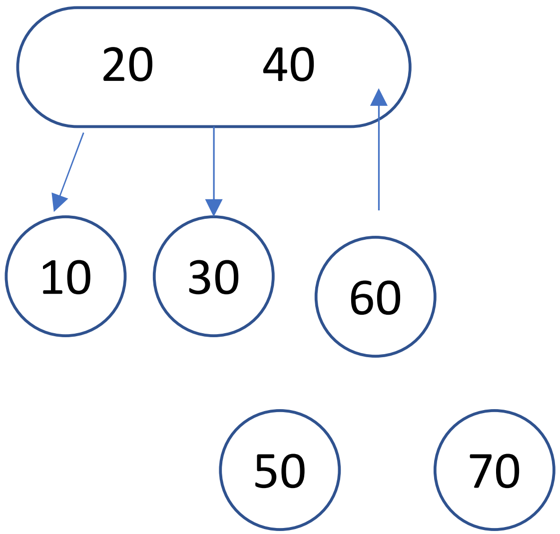
2-3 Trees

- Construct a 2-3 tree by inserting the following elements in the given order
  - → 10, 20, 30, 40, 50, 60, 70



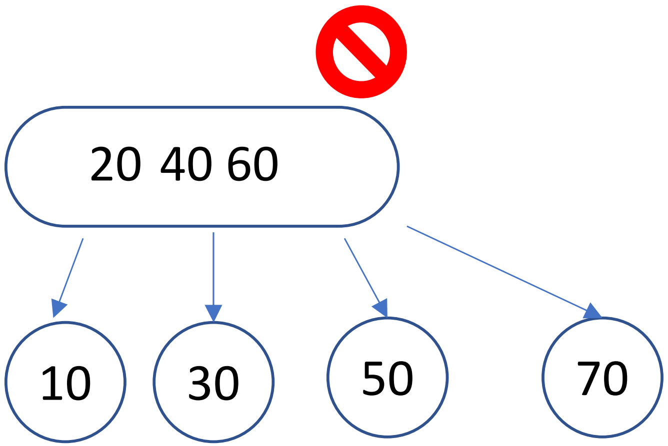
2-3 Trees

- Construct a 2-3 tree by inserting the following elements in the given order
  - → 10, 20, 30, 40, 50, 60, 70



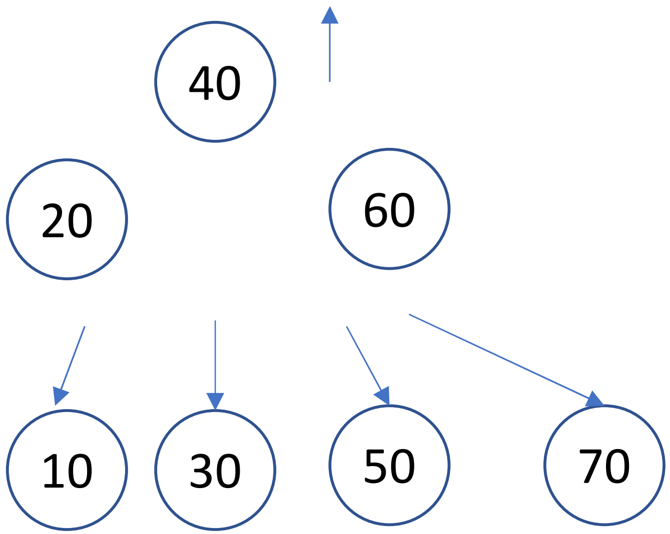
2-3 Trees

- Construct a 2-3 tree by inserting the following elements in the given order
  → 10, 20, 30, 40, 50, 60, 70



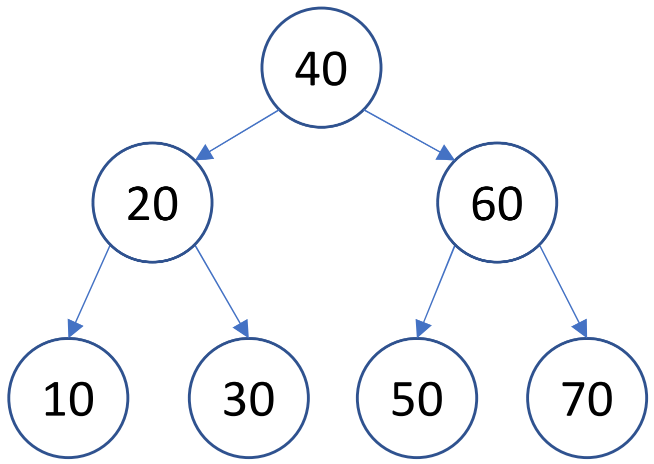
2-3 Trees

- Construct a 2-3 tree by inserting the following elements in the given order
  - → 10, 20, 30, 40, 50, 60, 70



2-3 Trees

- Construct a 2-3 tree by inserting the following elements in the given order
  - → 10, 20, 30, 40, 50, 60, 70



2-3 Trees

- 2-3 Trees grow downwards from the root when new nodes are inserted

    A. True

    B. False

Graph Concepts

- Construct a graph from the following

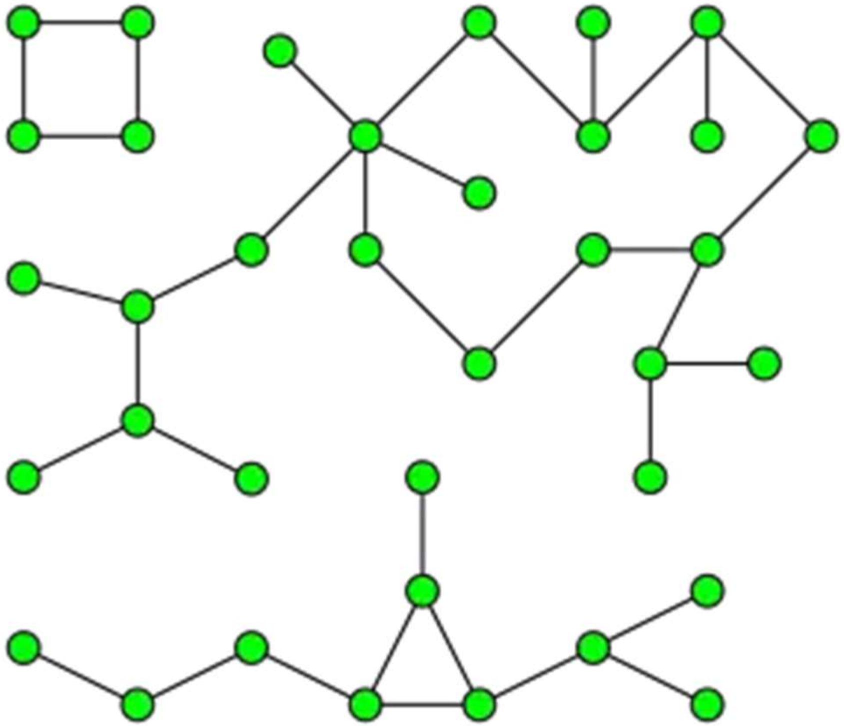
$|V|$ = 7 vertices V = {A, B, C, D, E, F, G}

$|E|$ = 9 edges E = {{A, B}, {A, D}, {A, E}, {B, C}, {B, D}, {B, E}, {C, E}, {C, F}, {D, E}}

Graph Concepts

- How many connected components are present in this graph?

A graph is connected if there exists a path between any two vertices



Connected component is a component of an undirected graph is a connected subgraph that is not part of any larger connected subgraph

Graph Concepts

- Which of the following statements are true?
  A. All trees are graphs
  B. All graphs are trees
  C. Both A and B
  D. None of the above

Graph Concepts

- Which of the following statements are true?
    A. All trees are graphs
    B. All graphs are trees
    C. Both A and B
    D. None of the above

A tree is an undirected graph in which any two vertices are connected by ***exactly one path***, or equivalently a connected acyclic undirected graph

Graph Concepts

Adjacency Matrix vs Adjacency List

Operations	Adjacency Matrix	Adjacency List
Storage Space	This representation makes use of VxV matrix, so space required is **$O(\|V\|^2)$**.	In this representation, for every vertex we store its neighbours. In the worst case, if a graph is connected O(V) is required for a vertex and O(E) is required for storing neighbours corresponding to every vertex .Thus, overall space complexity is O(|V|+|E|).
Adding a vertex	In order to add a new vertex to VxV matrix the storage must be increases to $(\|V\|+1)^2$. To achieve this, we need to copy the whole matrix. Therefore, the complexity is **$O(\|V\|^2)$**.	There are two pointers in adjacency list first points to the front node and the other one points to the rear node.Thus insertion of a vertex can be done directly in **O(1) time.**
Adding an edge	O(1)	O(1)
Removing a vertex	In order to remove a vertex from V*V matrix the storage must be decreased to $\|V\|^2$ from $(\|V\|+1)^2$. To achieve this, we need to copy the whole matrix. Therefore, the complexity is **$O(\|V\|^2)$**.	In order to remove a vertex, we need to search for the vertex which will require O(|V|) time in worst case, after this we need to traverse the edges and in worst case it will require O(|E|) time.Hence, total time complexity is **O(|V|+|E|)**.
Removing an edge	O(1)	To remove an edge traversing through the edges is required and in worst case we need to traverse through all the edges.Thus, the time complexity is **O(|E|)**.
Querying	O(1)	O(E/V) average O(V) worst case

Graph Concepts

- Which representation is better for representing a sparse graph?
  - A. Adjacency Matrix
  - B. Adjacency List

Graph Concepts

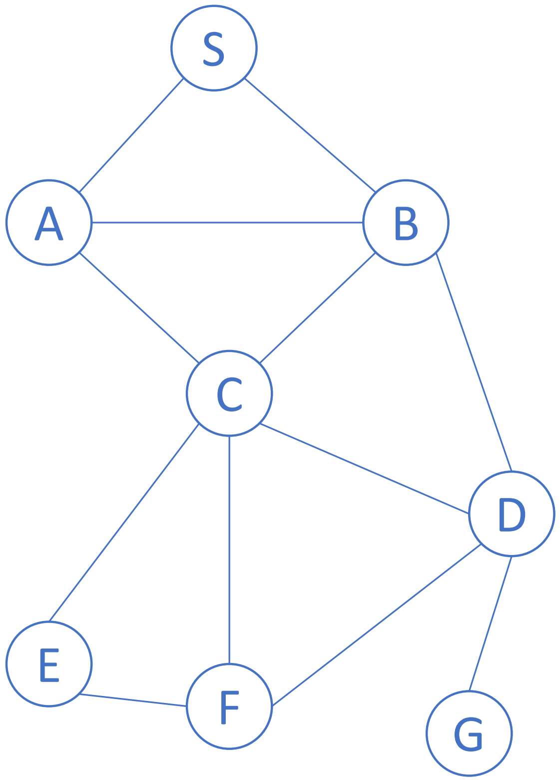
- Depth-first search of any graph always generates a unique tree

  A. True

  B. False

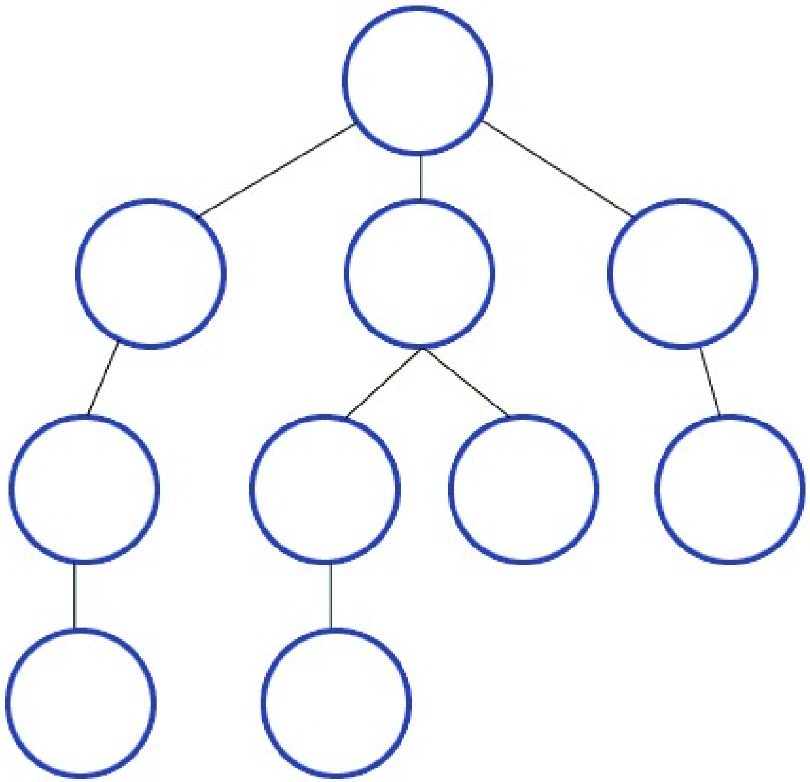
Graph Concepts

- Carry out DFS on this graph



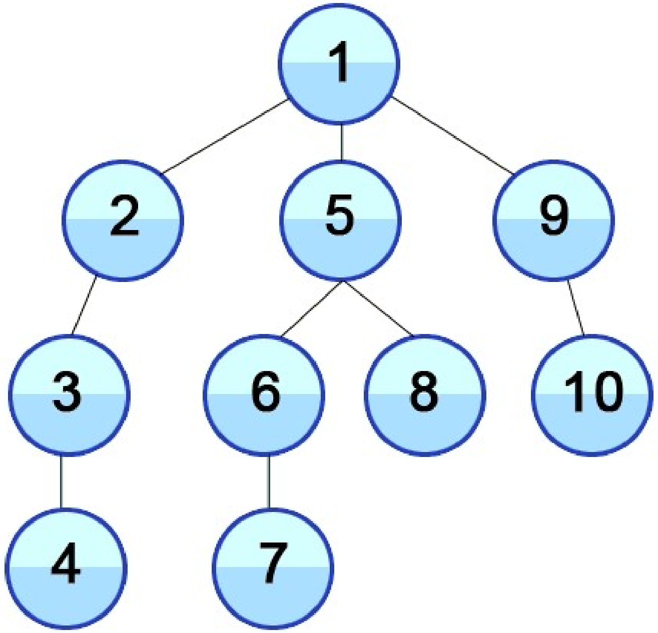
Graph Concepts

- Carry out DFS on this graph and number the nodes based on the visiting order



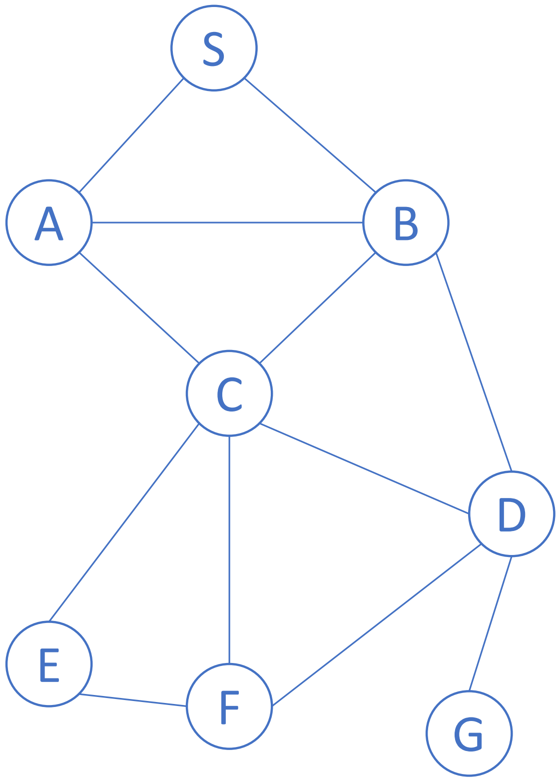
Graph Concepts

- Carry out DFS on this graph and number the nodes based on the visiting order



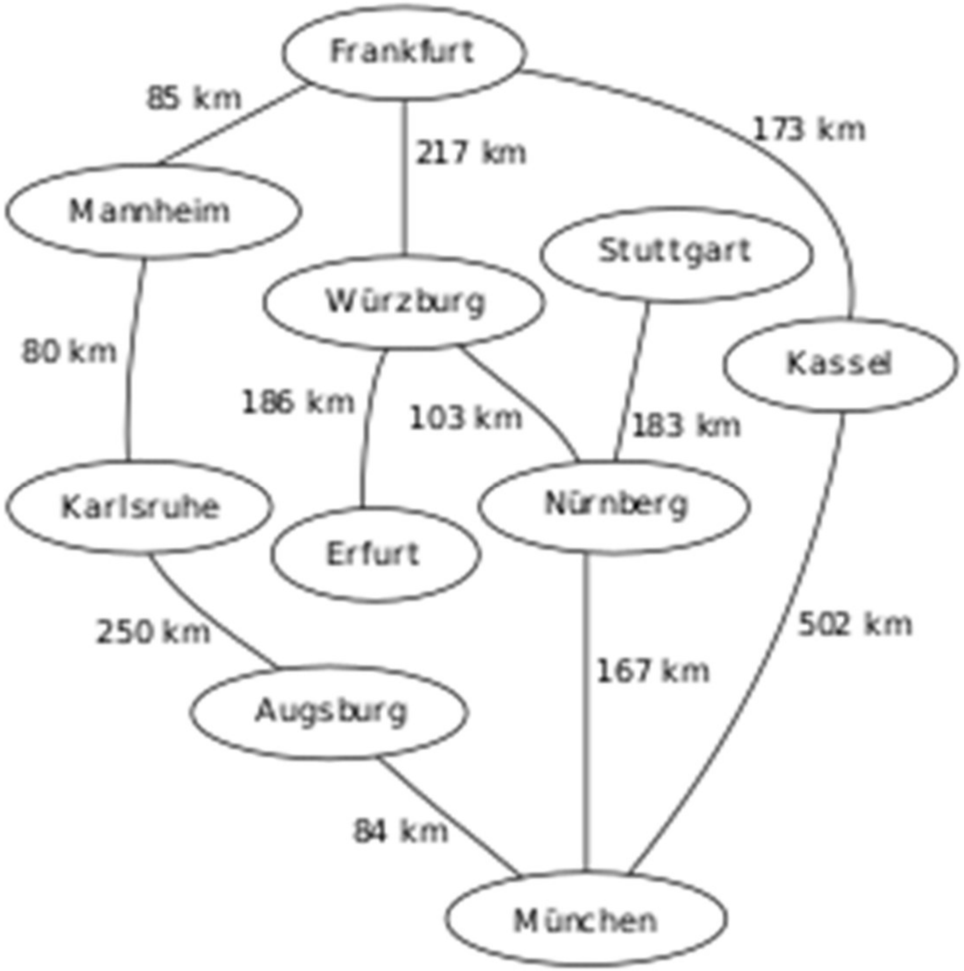
Graph Concepts

- Carry out BFS on this graph



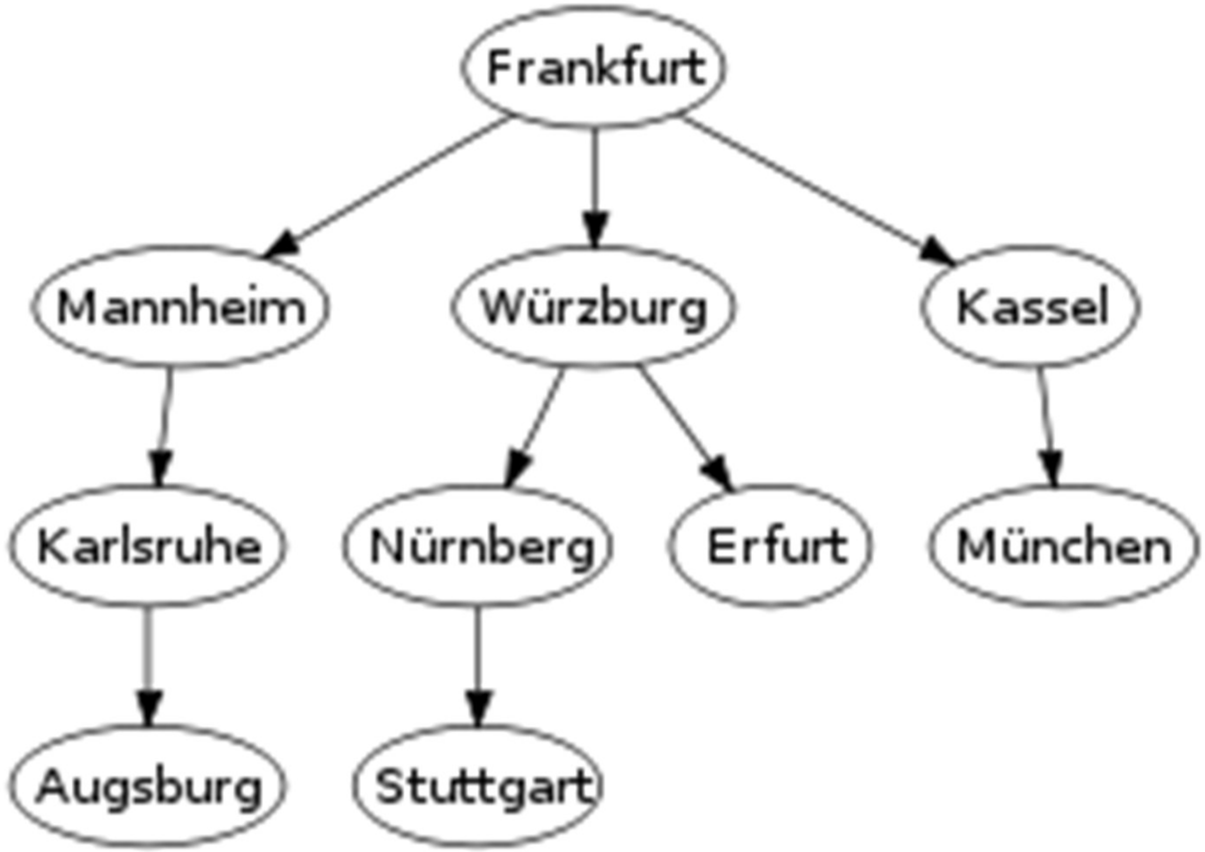
Graph Concepts

- Carry out BFS on this graph, starting at Frankfurt



Graph Concepts

- Carry out BFS on this graph, starting at Frankfurt



Graph Concepts

- Depth-first search can be used to find out the shortest distance from a node to all other nodes in an unweighted graph

  A. True

  B. False

Graph Concepts

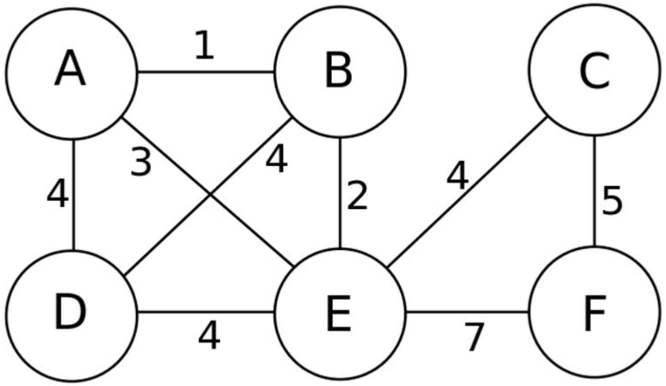
- Breadth-first search can be used to find out the shortest distance from a node to all other nodes in an unweighted graph

    A. True

    B. False

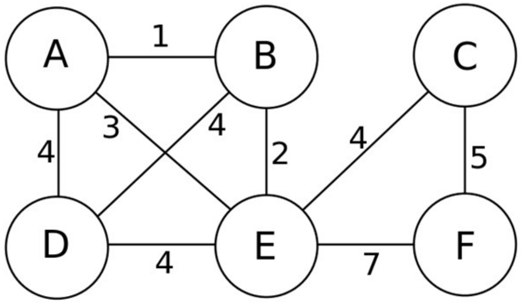
Graph Concepts

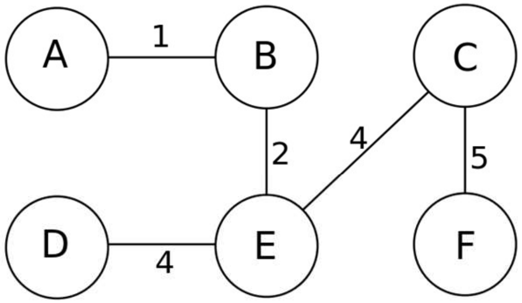
- Identify a minimum spanning tree of this graph

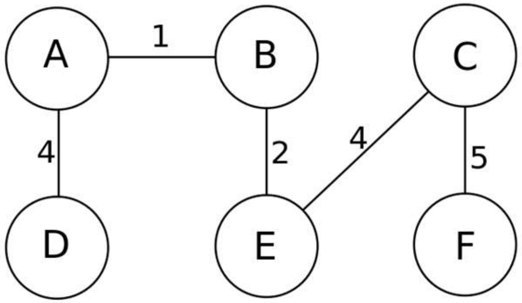


Graph Concepts

- Identify a minimum spanning tree of this graph

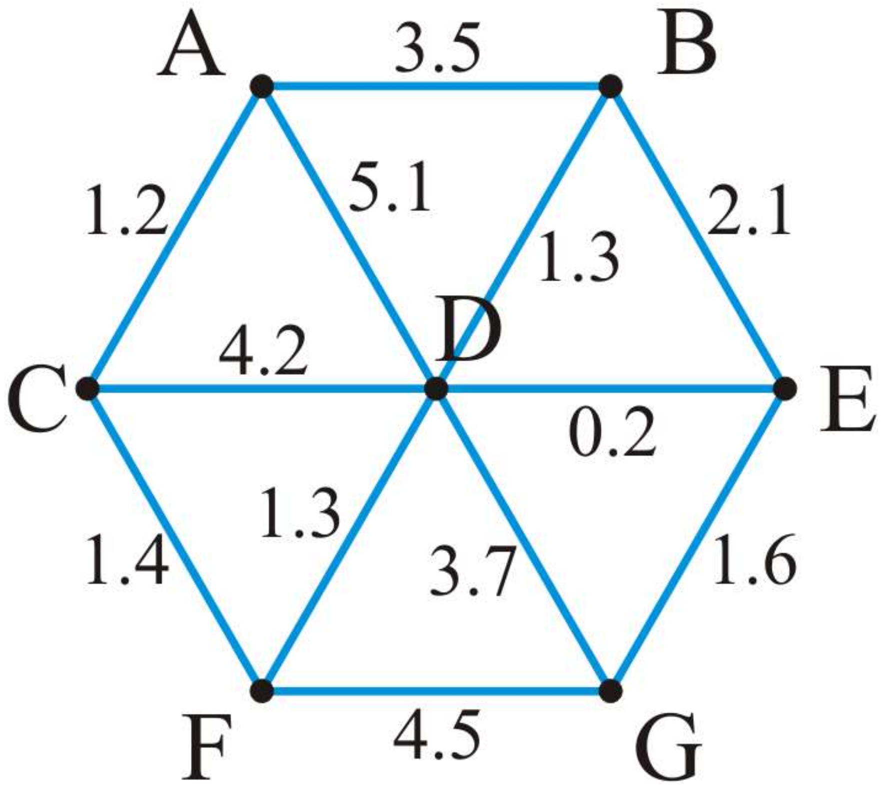






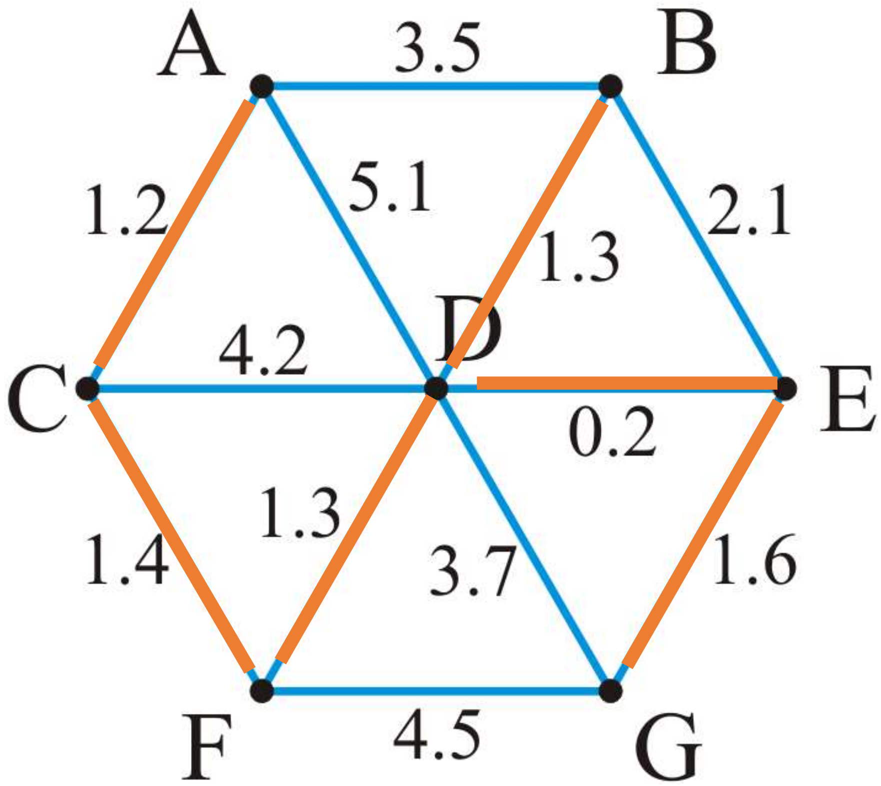
Graph Concepts

- Identify a minimum spanning tree of this graph



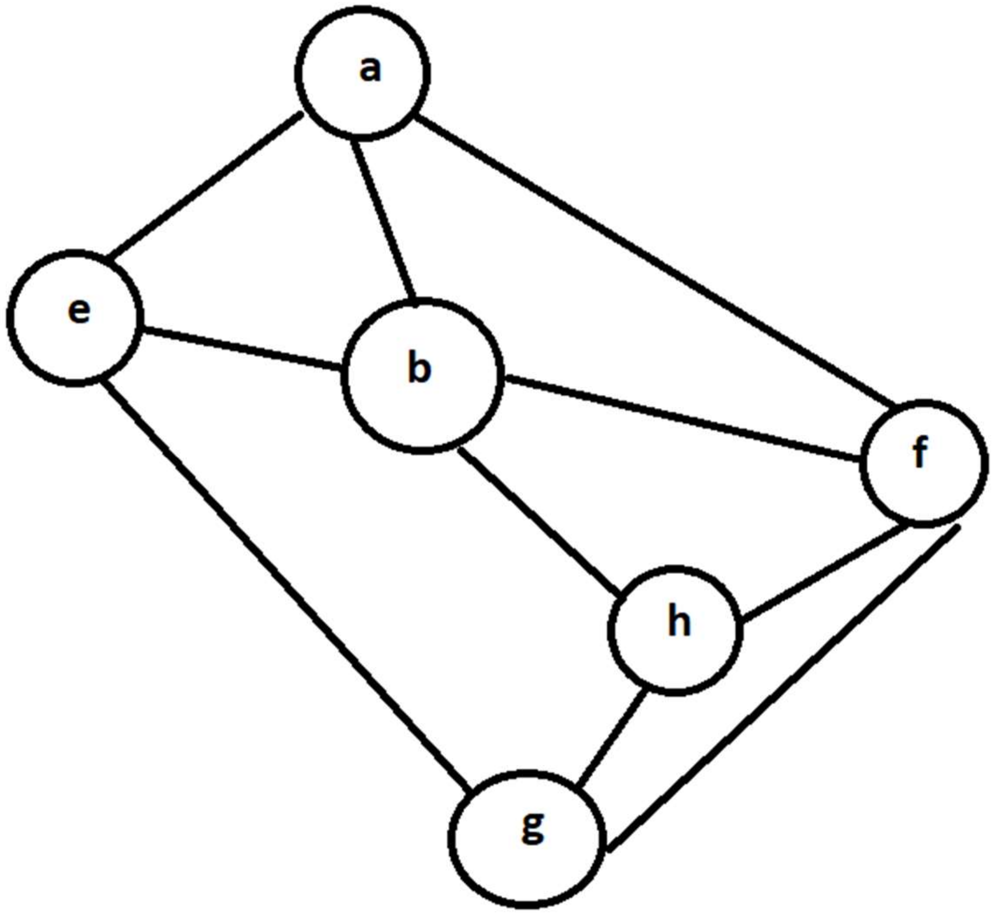
Graph Concepts

- Identify a minimum spanning tree of this graph



Graph Concepts

- Given this graph
- Among the following sequences:
  - (I) a b e g h f
  - (II) a b f e h g
  - (III) a b f h g e
  - (IV) a f g h b e



Which are the depth-first traversals of the above graph?

(A) I, II, and IV only

(B) I and IV only

(C) II, III, and IV only

(D) I, III, and IV only