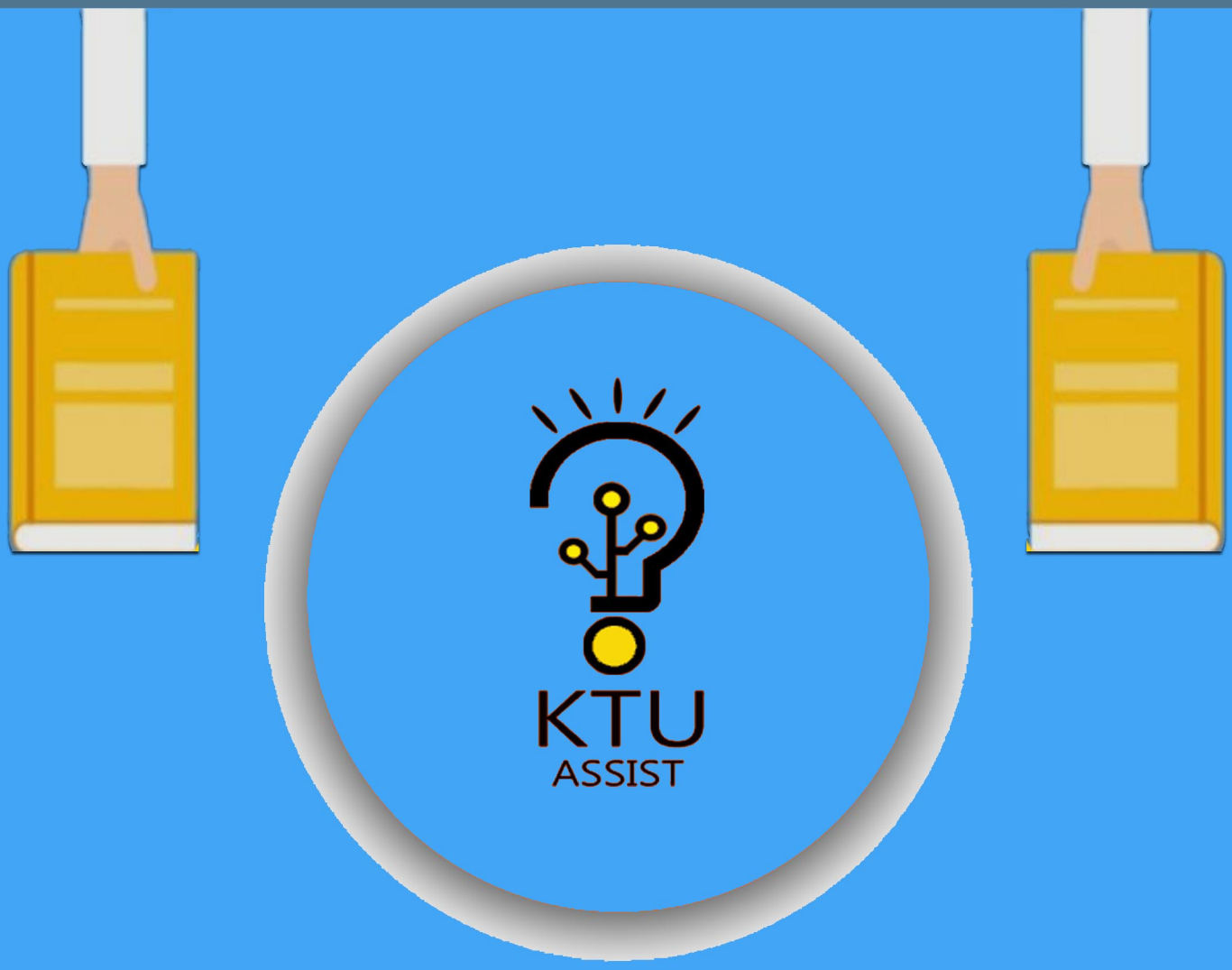
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

STUDY MATERIALS



a complete app for ktu students

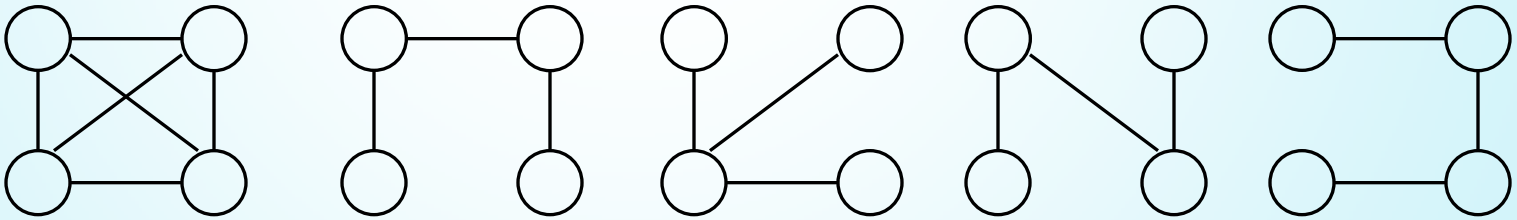
Get it on Google Play

www.ktuassist.in

Spanning Tree

Spanning trees

- Suppose you have a connected undirected graph
  - Connected: every node is reachable from every other node
  - Undirected: edges do not have an associated direction
- spanning tree of the graph is a connected subgraph in which there are no cycles

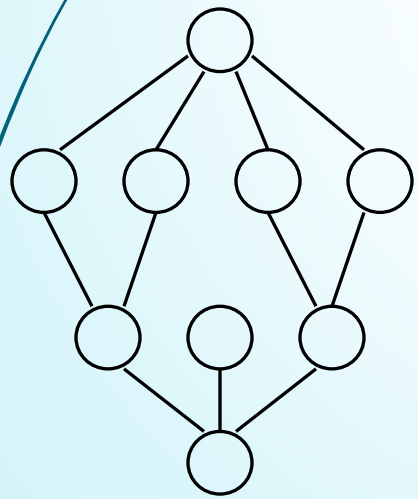


A connected, undirected graph

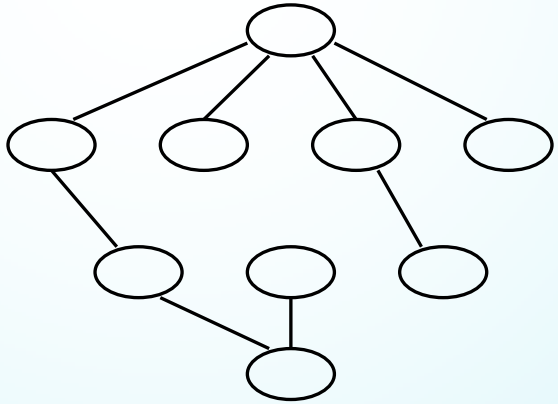
Four of the spanning trees of the graph

Finding a spanning tree

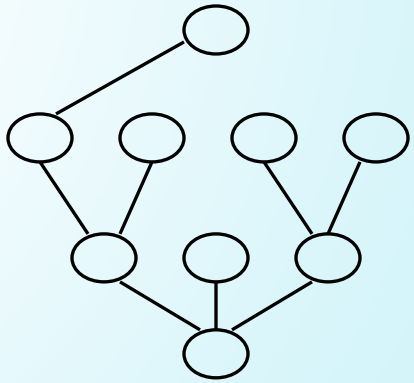
- pick an initial node and call it part of the spanning tree
- do a search from the initial node:
  - each time you find a node that is not in the spanning tree, add to the spanning tree both the new node *and* the edge you followed to get to it



An undirected graph



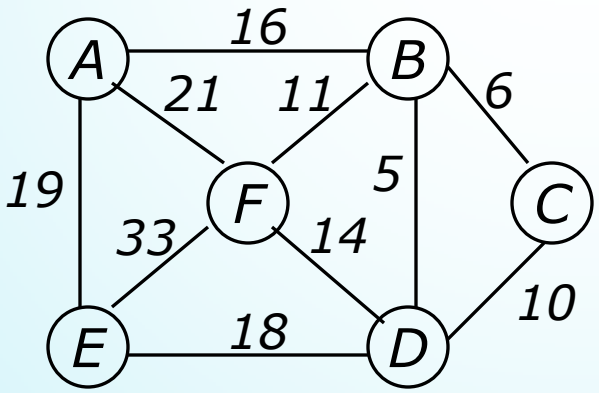
One possible result of a BFS starting from top



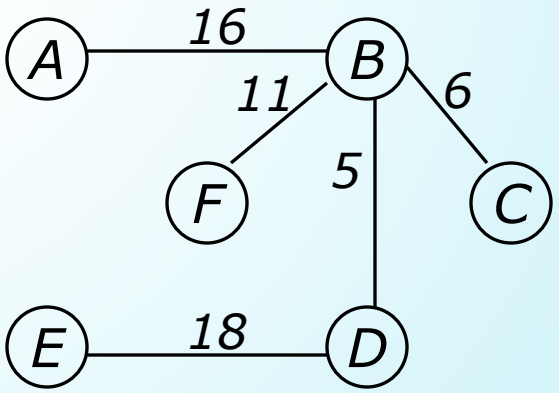
One possible result of a DFS starting from top

Minimum-cost spanning trees

- Suppose you have a connected undirected graph with a weight (or cost) associated with each edge
- Cost of a spanning tree would be sum of the costs of its edges
- A minimum-cost spanning tree is a spanning tree that has the lowest cost



A connected, undirected graph



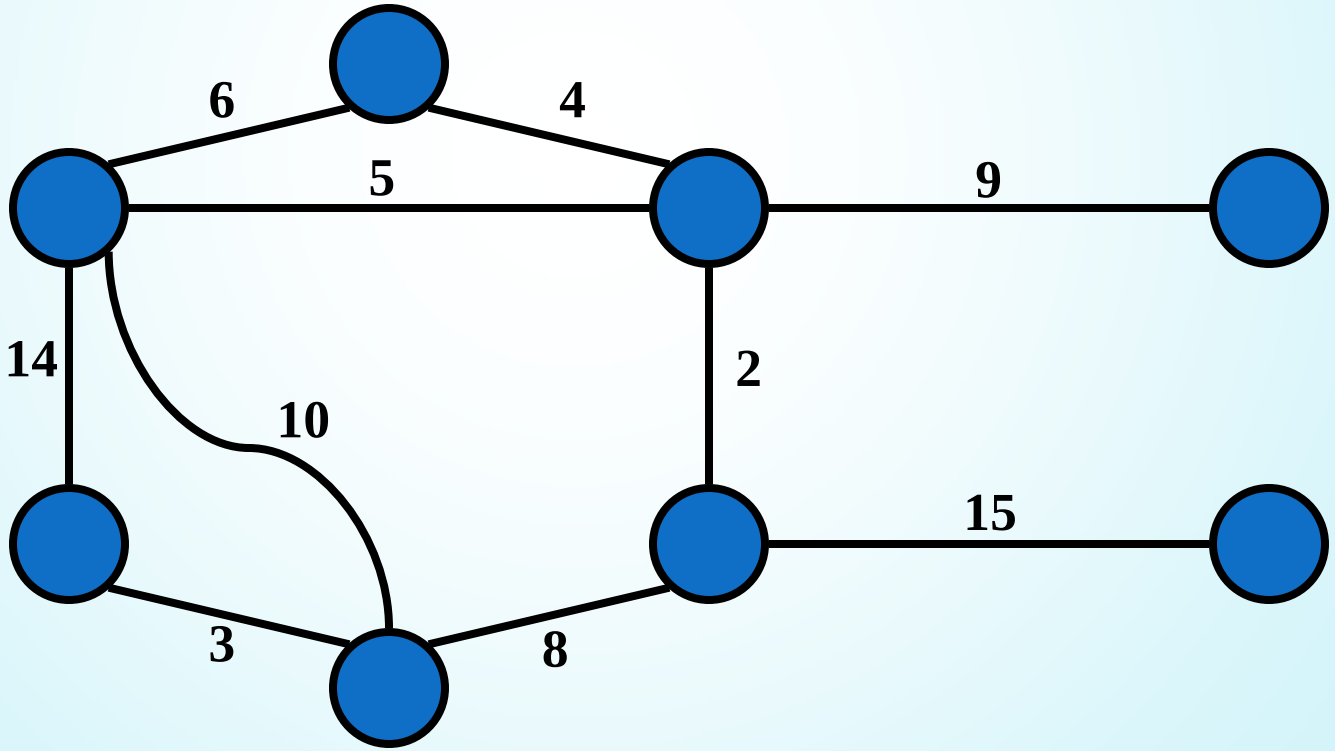
A minimum-cost spanning tree

Minimizing Costs

- Suppose you want to supply a set of houses (say, in a new subdivision) with:
  - electric power
  - water
  - sewage lines
  - telephone lines

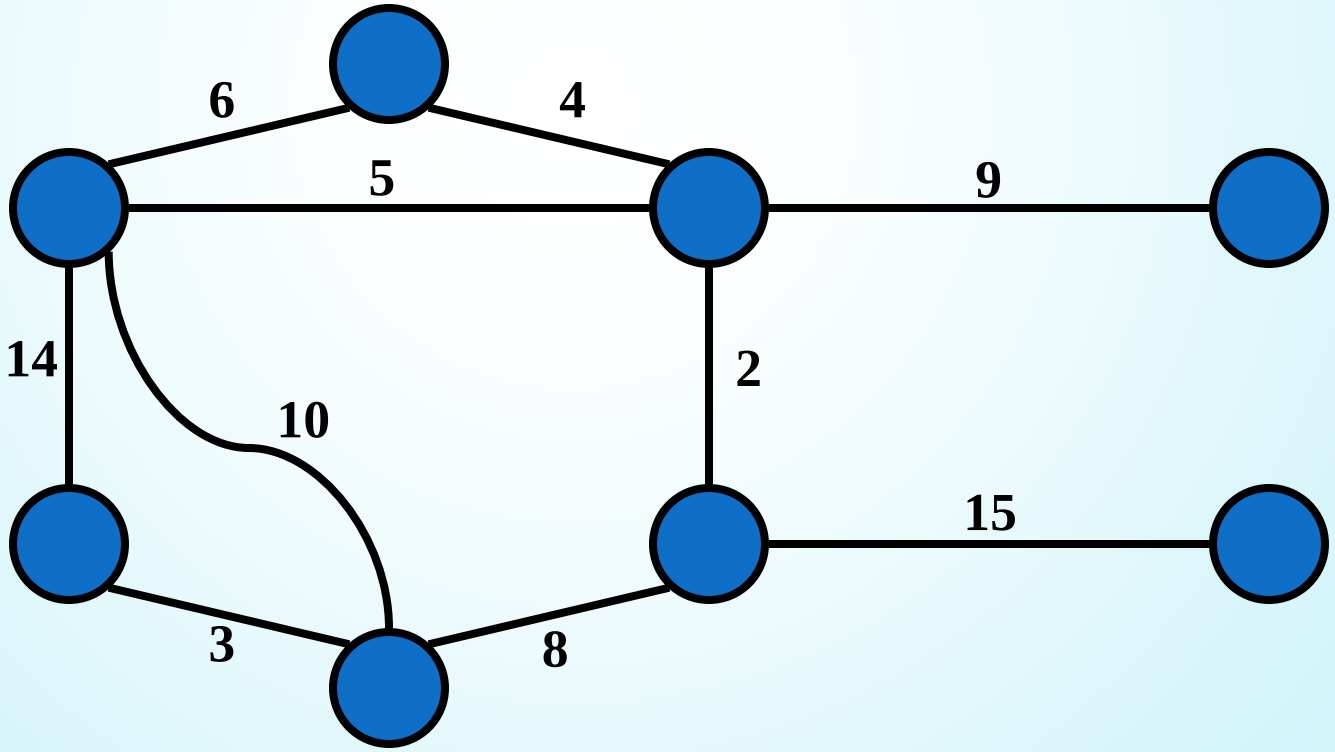
Minimum Spanning Tree

- Problem: given a connected, undirected, weighted graph:



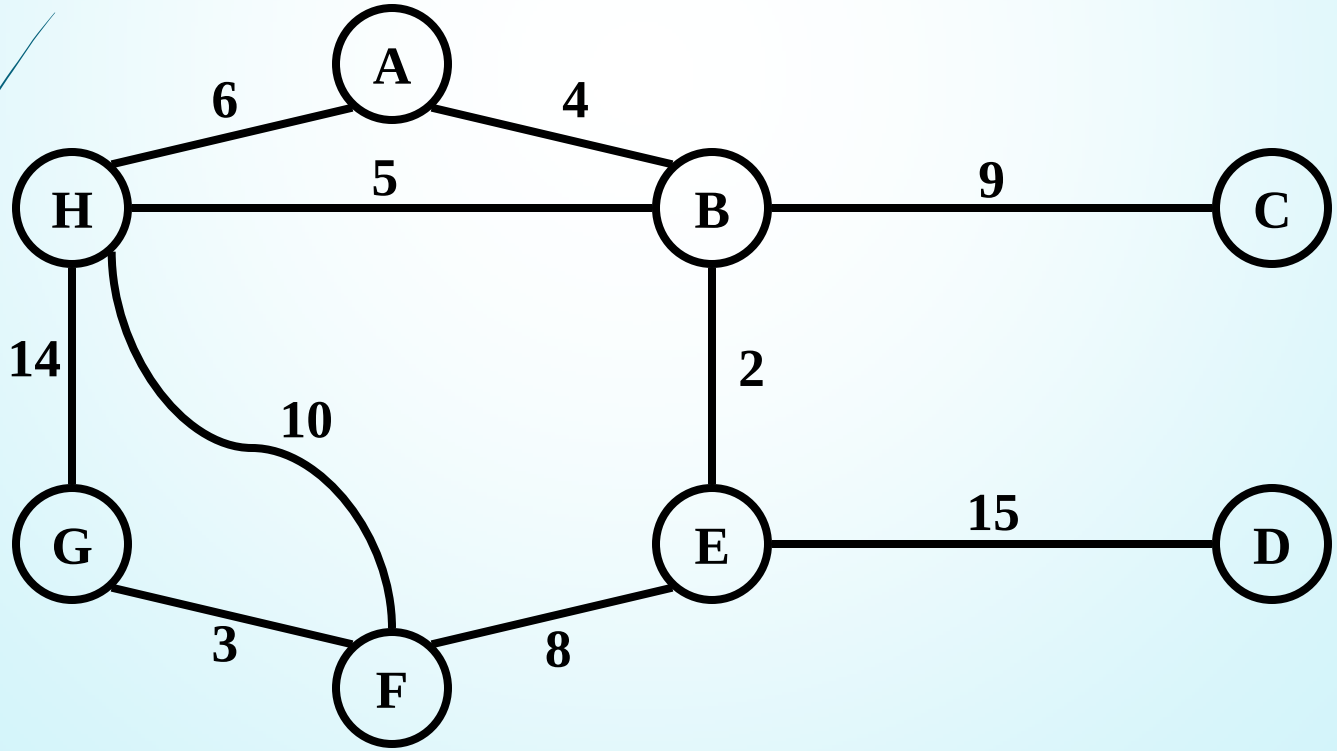
Minimum Spanning Tree

- Problem: given a connected, undirected, weighted graph, find a *spanning tree* using edges that minimize the total weight



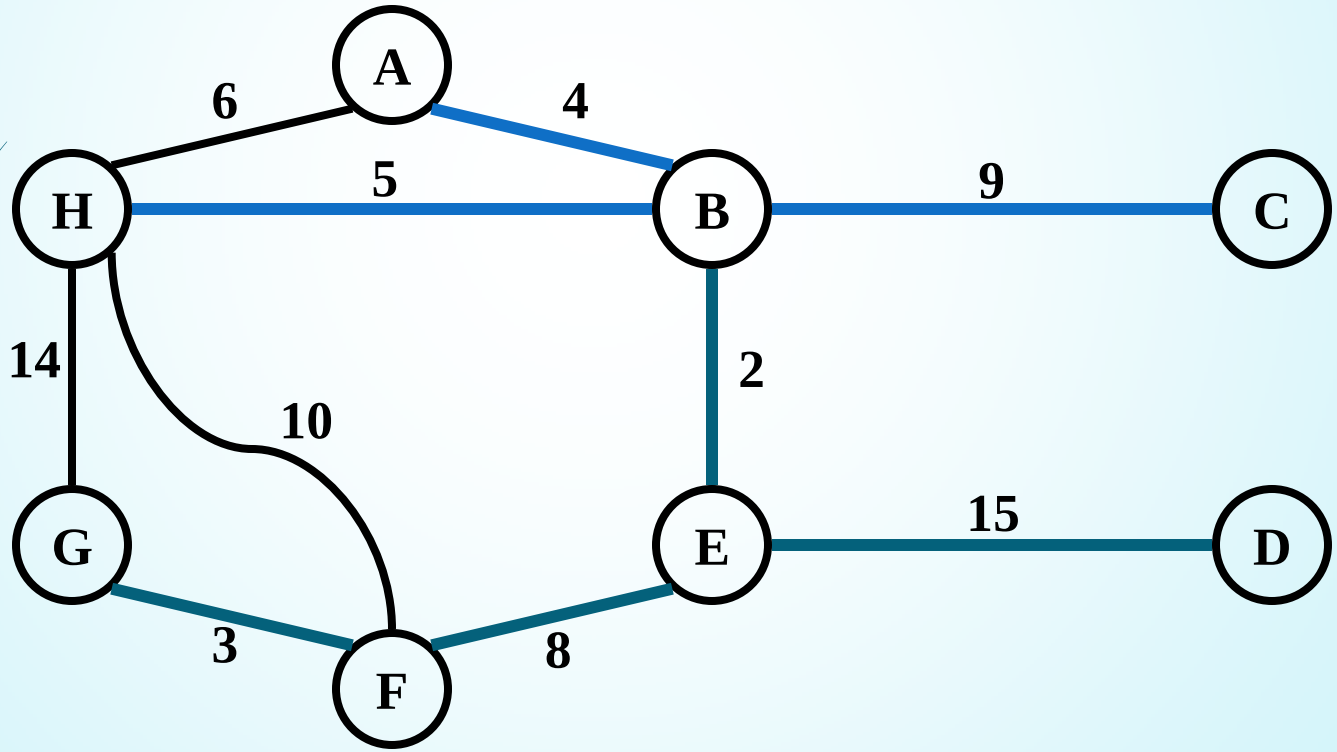
Minimum Spanning Tree

- *Which edges form the minimum spanning tree (MST) of the below graph?*



Minimum Spanning Tree

- Answer:



Finding spanning trees

- There are two basic algorithms for finding minimum-cost spanning trees, and both are greedy algorithms
- **Kruskal's algorithm:** Start with *no* nodes or edges in the spanning tree, and repeatedly add the cheapest edge that does not create a cycle
- **Prim's algorithm:** Start with any *one node* in the spanning tree, and repeatedly add the cheapest edge, and the node it leads to, for which the node is not already in the spanning tree.

END

facebook.com/ktuassist instagram.com/ktu_assist