Greedy Technique

Kruskal's algorithm

Kruskal’s algorithm

- Joseph Kruskal (second year graduate student) – paper appeared in **Proceedings of the American Mathematical Society, pp. 48–50 in 1956**

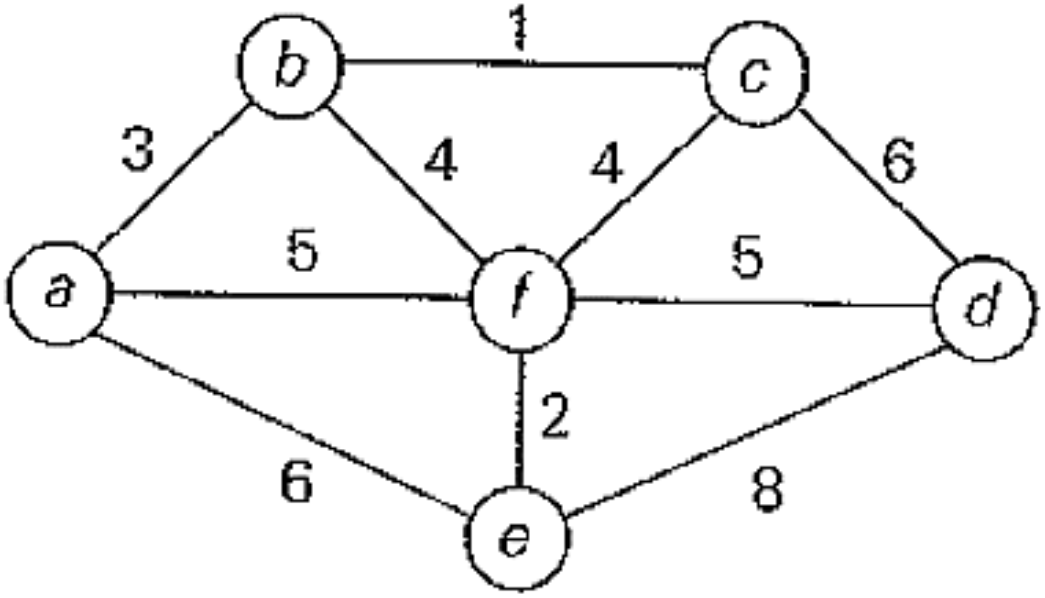
- Uses greedy approach

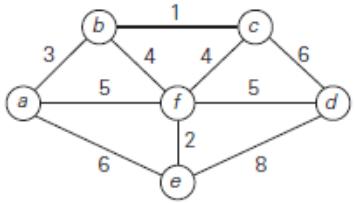
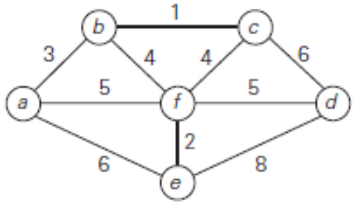
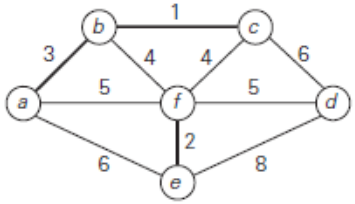
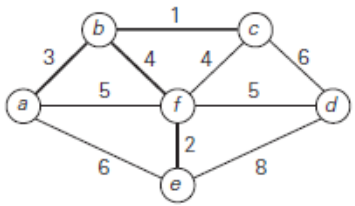
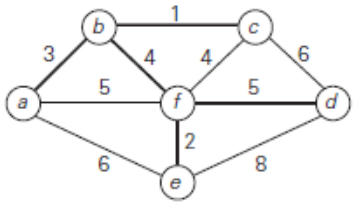
- Finds MST
  - Prim's algorithm,
  - reverse-delete algorithm,
  - Borůvka's algorithm

ALGORITHM *Kruskal*(G)

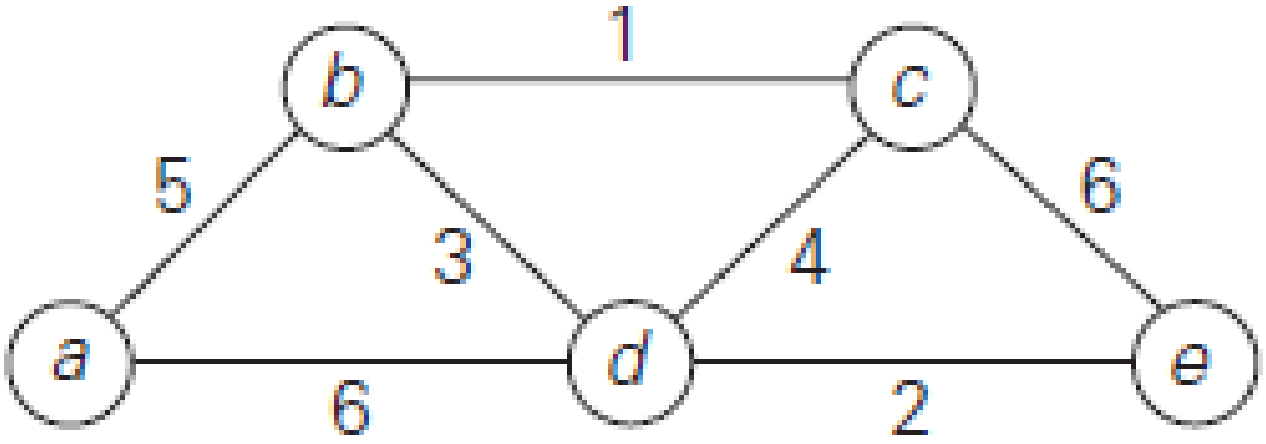
//Kruskal's algorithm for constructing a minimum spanning tree
//Input: A weighted connected graph $G = \langle V, E \rangle$
//Output: E_T, the set of edges composing a minimum spanning tree of G
sort E in nondecreasing order of the edge weights $w(e_{i_1}) \leq \cdots \leq w(e_{i_{|E|}})$
$E_T \leftarrow \varnothing$; $ecounter \leftarrow 0$ //initialize the set of tree edges and its size
$k \leftarrow 0$ //initialize the number of processed edges
while $ecounter < |V| - 1$ **do**
    $k \leftarrow k + 1$
    if $E_T \cup \{e_{i_k}\}$ is acyclic
        $E_T \leftarrow E_T \cup \{e_{i_k}\}$; $ecounter \leftarrow ecounter + 1$
return E_T

Apply Kruskal's algorithm to find MST



Tree edges	Sorted list of edges	Illustration
	bc ef ab bf cf af df ae cd de 1 2 3 4 4 5 5 6 6 8	
bc 1	bc **ef** ab bf cf af df ae cd de 1 2 3 4 4 5 5 6 6 8	
ef 2	bc ef **ab** bf cf af df ae cd de 1 2 3 4 4 5 5 6 6 8	
ab 3	bc ef ab **bf** cf af df ae cd de 1 2 3 4 4 5 5 6 6 8	
bf 4	bc ef ab bf cf af **df** ae cd de 1 2 3 4 4 5 5 6 6 8	
df 5		

Apply Kruskal's algorithm to find MST



Prims Algorithm	**Kruskal Algorithm**
It start to build the MST from any of the Node.	It start to build the MST from Minimum weighted vertex in the graph.
Adjencary Matrix , Binary Heap or Fibonacci Heap is used in Prims algorithm	Disjoint Set is used in Kruskal Algorithm.
Prims Algorithm run faster in dense graphs	Kruskal Algorithm run faster in sparse graphs
Time Complexity is O(EV log V) with binay heap and O(E+V log V) with fibonacci heap.	Time Complexity is **O(E log V)**
The next Node included must be connected with the node we traverse	The next edge include may or may not be connected but should not form the cycle.
It traverse the one node saveral time in order to get it minimum distance	It travese the edge only once and based on cycle it will either reject it or accept it,
Greedy Algorithm	Greedy Algorithm