

# Greedy

- Always Efficient
- Does not guarantee the optimal solution
- Makes the best choice at the <sup>current</sup> moment, without considering of its later outcome
- Only applicable on optimization problems
- Optimal substructure

- ① Activity selection problem
- ② Fractional knapsack

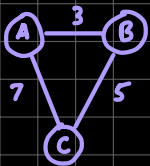
## Minimum spanning Tree (MST)

- MST is a subset of edges from a weighted, connected, undirected graph that connects all vertices together without any cycles, and with the lowest possible total edge weight

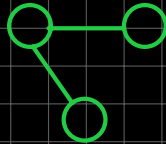
### Edge Weight:

- Sum of cost, distance, capacity or any other metric b/w 2 vertices the edge connects

Q-

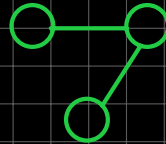


Spanning trees:



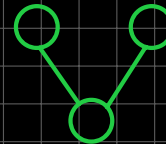
Edge Weight:

10



8

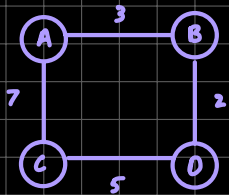
MST



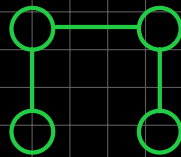
12

- connect all vertices without forming any cycles

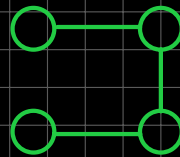
Q-



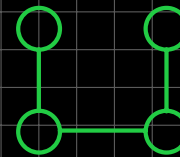
Spanning Trees:



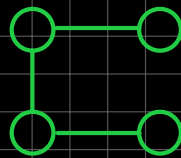
12



10  
= MST



14



15

- No. of Edges =  $|V| - 1$

- Generic Algo:
- create set A of edges (representing MST)
- Incrementally add edges to A (add only safe edge)

### Safe Edge:

- An edge  $(u, v)$  is safe for A iff  $A \cup (u, v) \subseteq$  of some MST

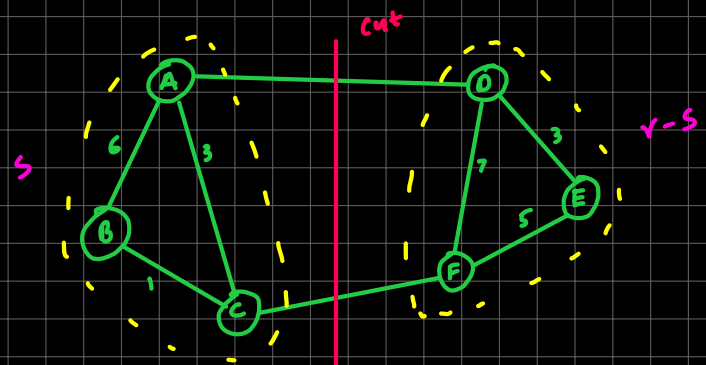
• ALGO B50k:

$A \leftarrow \emptyset$

while A is not a spanning Tree  
  find a safe edge  $(u,v)$  for A  
   $A \leftarrow A \cup \{(u,v)\}$

return A

• Finding Safe edge



• No edge in A crosses this division

Minimum edge

• Light weight edge chosen which is cut by the line

• Light weight edge is the safe edge

Prim's Algo