

# Minimum Spanning Tree

MD. MUKTAR HOSSAIN

LECTURER

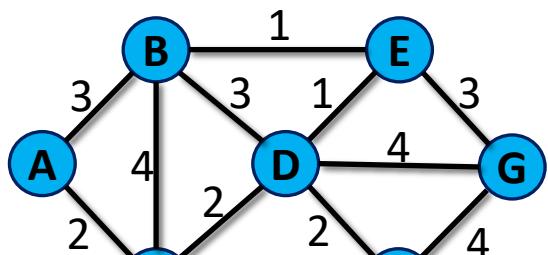
---

DEPT. OF CSE

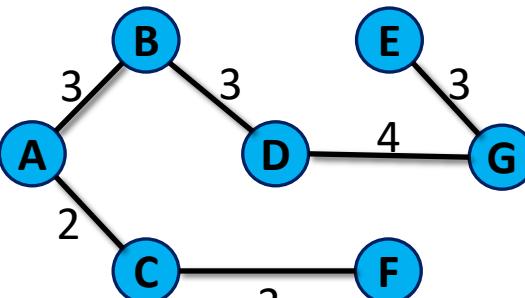
VARENDRA UNIVERSITY

# Spanning Tree

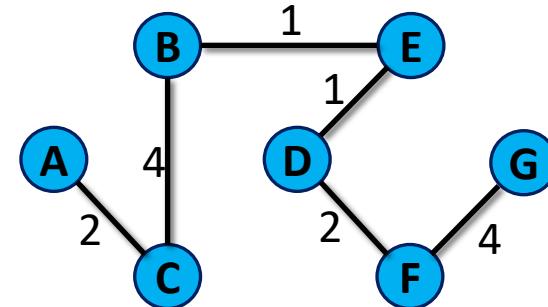
A *spanning tree* of a graph is a subgraph that includes all the vertices of the original graph and is also a tree (a *connected acyclic graph*).



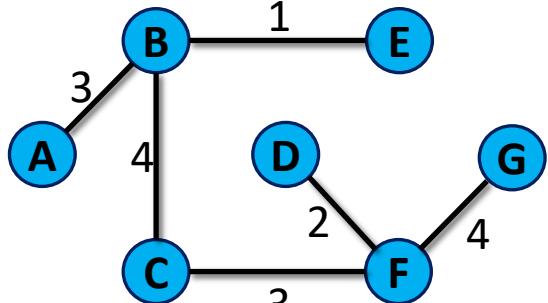
Graph



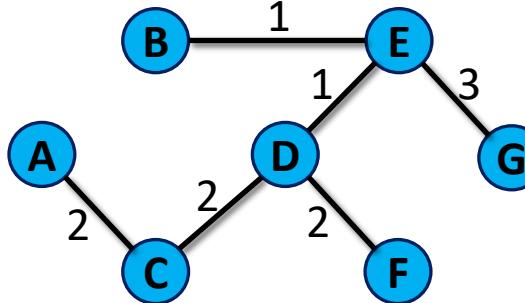
ST-1



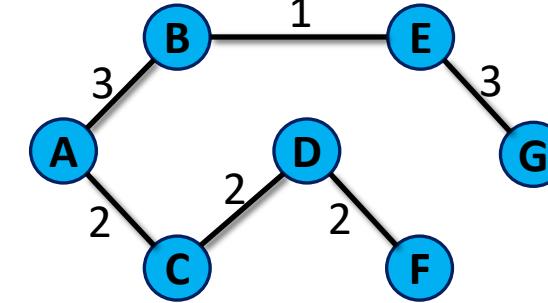
ST-2



ST-3



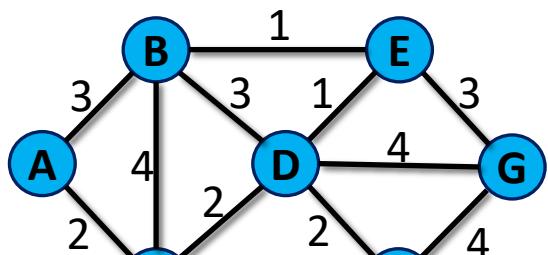
ST-4



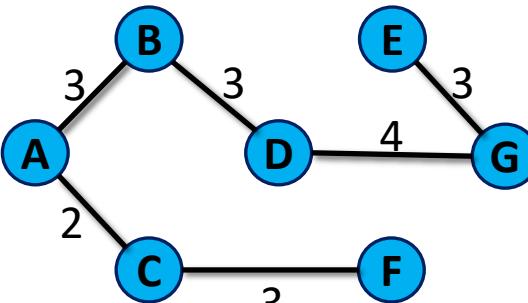
ST-5

# Minimum Spanning Tree

A **minimum spanning tree (MST)** is defined as a spanning tree that has the **minimum weight among all the possible spanning trees**.

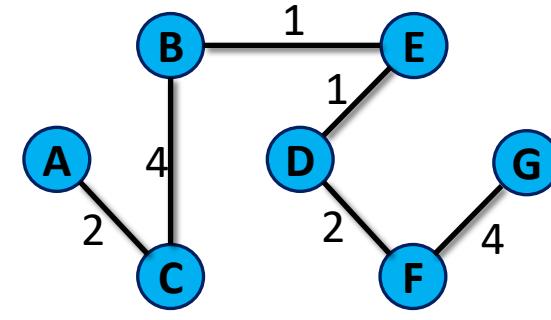


Graph



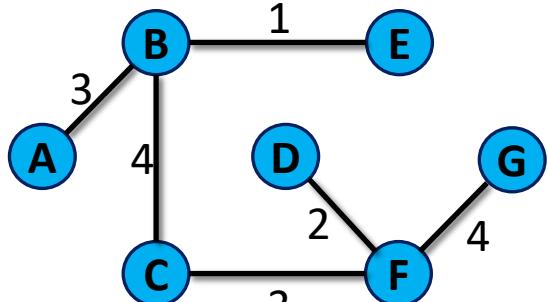
ST-1

18



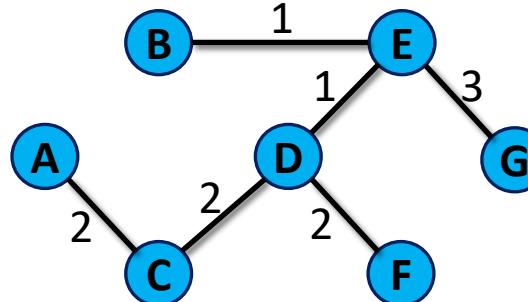
ST-2

14



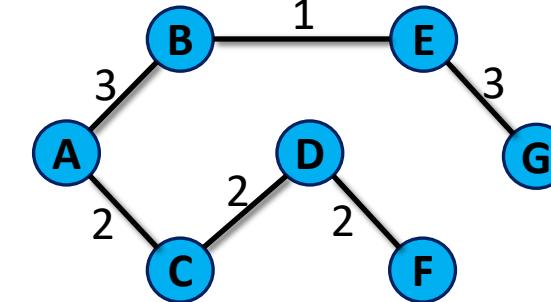
ST-3

17



ST-4

11



ST-5

13

# MST Applications

---

## □ Computer Networks

- ✓ Connect computers/routers with minimum cabling cost.
- ✓ Avoids cycles, ensures full connectivity.

## □ Telecommunication Networks

- ✓ Design cost-effective layouts for telephone lines and internet infrastructure.

## □ Road and Railway Systems

- ✓ Connect cities or regions with minimal construction cost.
- ✓ Ideal for planning roadways, pipelines, and railway tracks.

## □ Electrical Grid Design

- ✓ Distribute power with minimum wiring.
- ✓ Reduces construction and energy transmission costs.

## □ Game Map Design

- ✓ Connect rooms/areas with minimal paths in procedurally generated maps.

## □ Traveling Salesman Problem (TSP) Approximation

- ✓ MST is used in heuristics for finding near-optimal TSP tours.

## □ Wireless Sensor Networks

- ✓ Minimizes energy usage by reducing total communication path cost.

## □ Clustering in Machine Learning

- ✓ Used in hierarchical (single-linkage) clustering.
- ✓ Groups similar data points by cutting longest edges in the MST.

# Minimum Spanning Tree

---

## Requirements

- The graph must be connected and weighted (edges have weights).
- There can be more than one MST if multiple trees have the same total weight.

## Key Properties

- Total edges =  $V - 1$
- Optimizes cost, such as minimizing the length of wire needed to connect all computers in a network.

## Algorithms to find Minimum Spanning Tree

- Kruskal's Minimum Spanning Tree Algorithm
- Prim's Minimum Spanning Tree Algorithm

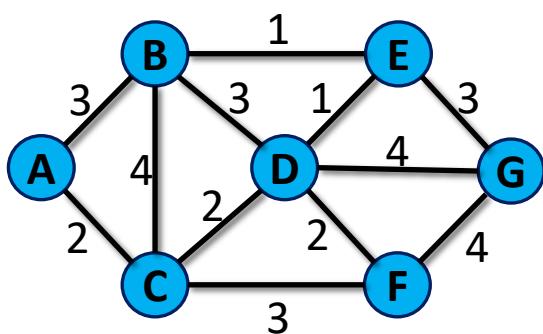
# Kruskal's MST Algorithm

---

## Algorithm

1. Sort all the edges in a ascending or non-decreasing order of their weight.
2. Pick the smallest edge. Check if it forms a cycle with the spanning tree formed so far.
  - i. If the cycle is not formed, include this edge.
  - ii. Else, discard it.
3. Repeat step 2 until there are  $(V-1)$  edges in the spanning tree.

# Kruskal's MST Algorithm

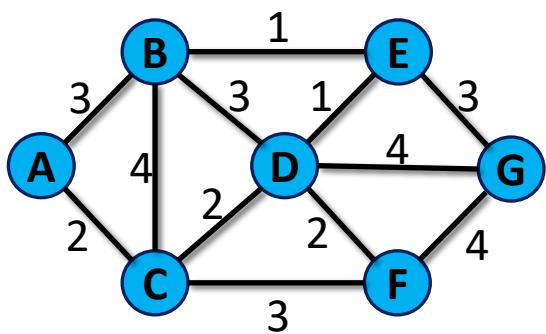


Edges	Cost
AB	3
AC	2
BC	4
BE	1
BD	3
CD	2
CF	3
DE	1
DF	2
DG	4
EG	3
FG	4

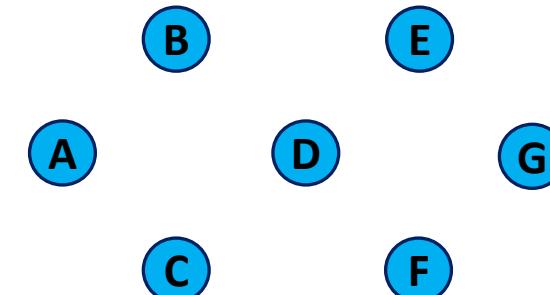
Sort by Cost

Edges	Cost
BE	1
DE	1
AC	2
CD	2
DF	2
AB	3
BD	3
CF	3
EG	3
BC	4
DG	4
FG	4

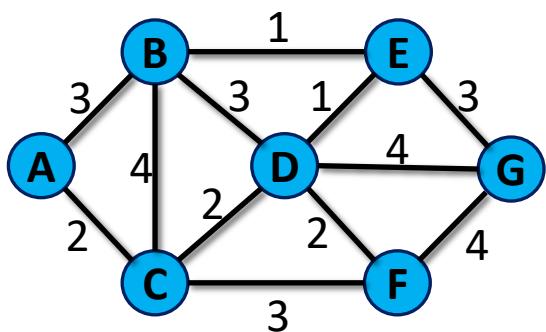
# Kruskal's MST Algorithm



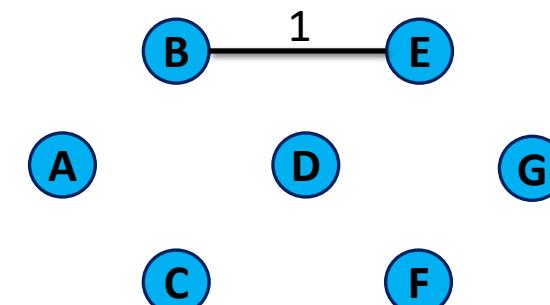
Edges	Cost
BE	1
DE	1
AC	2
CD	2
DF	2
AB	3
BD	3
CF	3
EG	3
BC	4
DG	4
FG	4



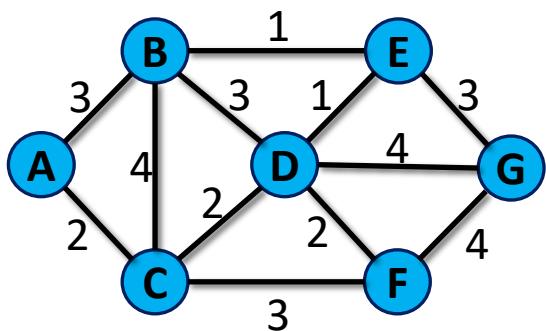
# Kruskal's MST Algorithm



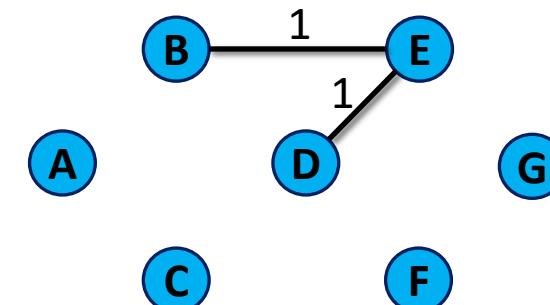
Edges	Cost
BE	1
DE	1
AC	2
CD	2
DF	2
AB	3
BD	3
CF	3
EG	3
BC	4
DG	4
FG	4



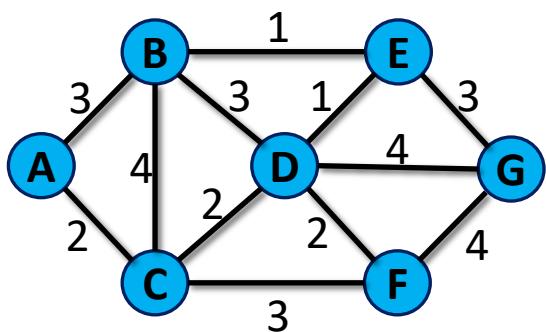
# Kruskal's MST Algorithm



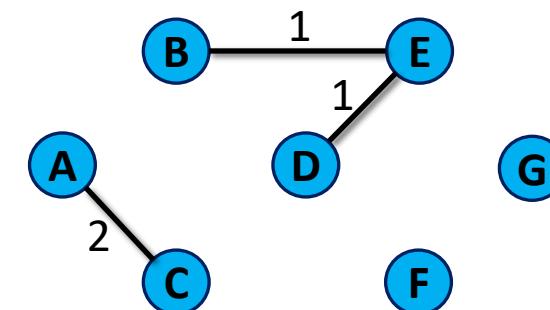
Edges	Cost
BE	1
DE	1
AC	2
CD	2
DF	2
AB	3
BD	3
CF	3
EG	3
BC	4
DG	4
FG	4



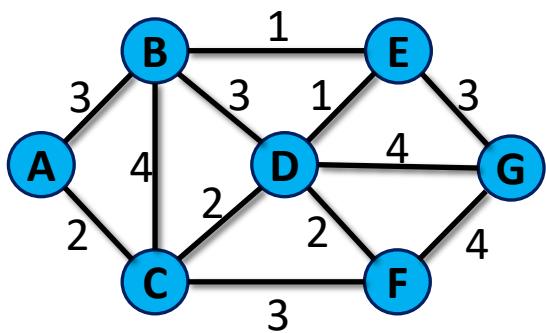
# Kruskal's MST Algorithm



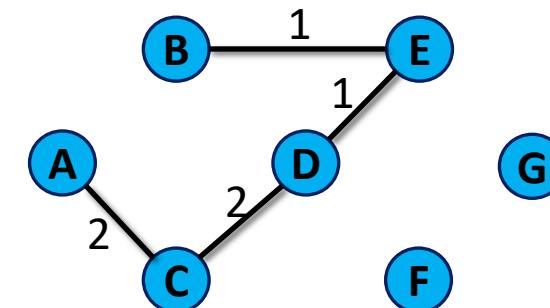
Edges	Cost
BE	1
DE	1
AC	2
CD	2
DF	2
AB	3
BD	3
CF	3
EG	3
BC	4
DG	4
FG	4



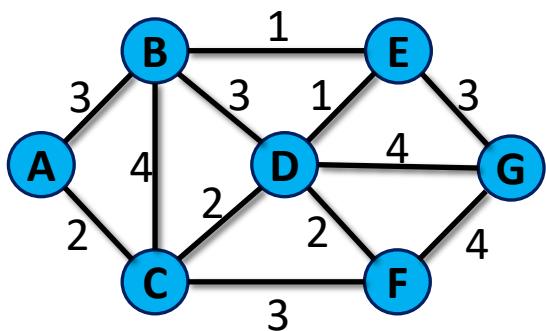
# Kruskal's MST Algorithm



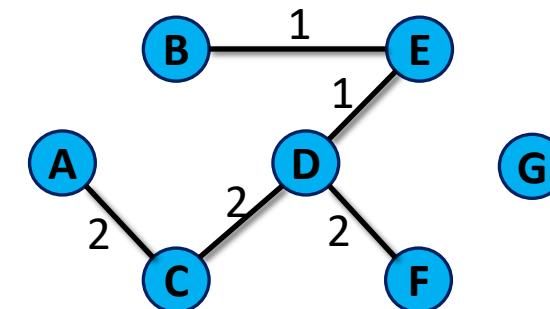
Edges	Cost
BE	1
DE	1
AC	2
CD	2
DF	2
AB	3
BD	3
CF	3
EG	3
BC	4
DG	4
FG	4



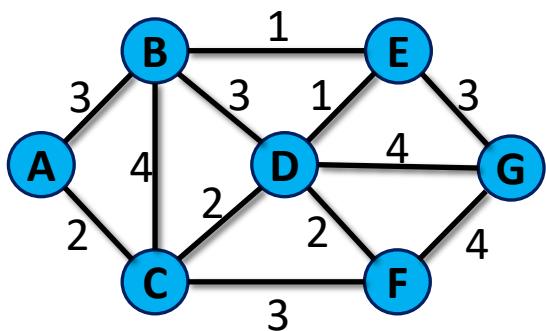
# Kruskal's MST Algorithm



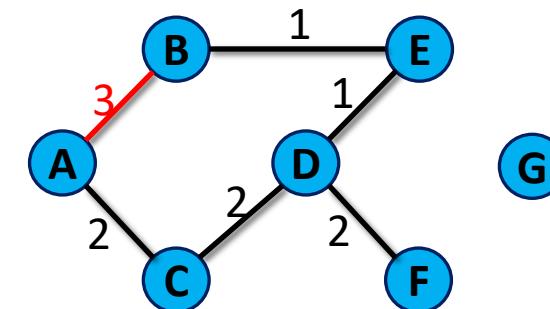
Edges	Cost
BE	1
DE	1
AC	2
CD	2
DF	2
AB	3
BD	3
CF	3
EG	3
BC	4
DG	4
FG	4



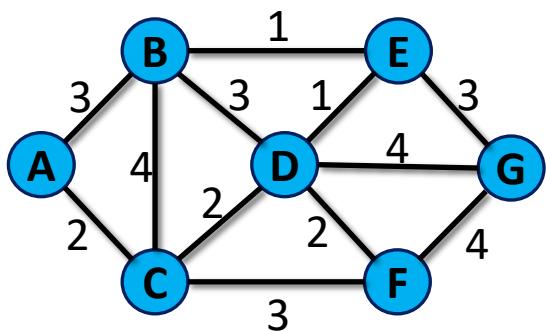
# Kruskal's MST Algorithm



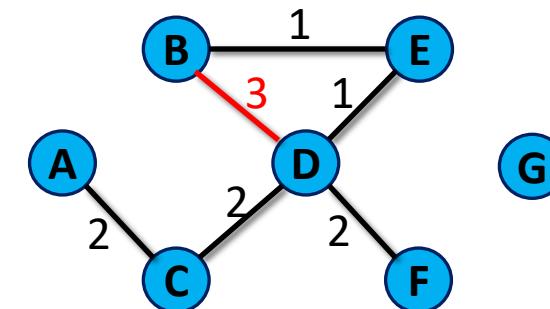
Edges	Cost
BE	1
DE	1
AC	2
CD	2
DF	2
AB	3
BD	3
CF	3
EG	3
BC	4
DG	4
FG	4



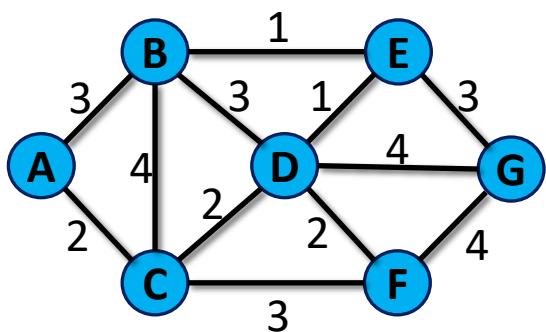
# Kruskal's MST Algorithm



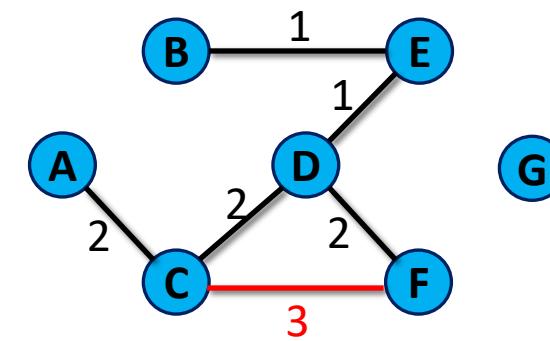
Edges	Cost
BE	1
DE	1
AC	2
CD	2
DF	2
AB	3
BD	3
CF	3
EG	3
BC	4
DG	4
FG	4



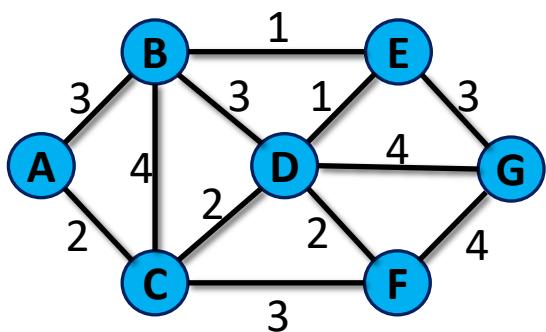
# Kruskal's MST Algorithm



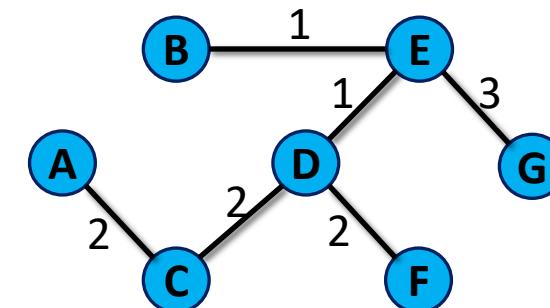
Edges	Cost
BE	1
DE	1
AC	2
CD	2
DF	2
AB	3
BD	3
CF	3
EG	3
BC	4
DG	4
FG	4



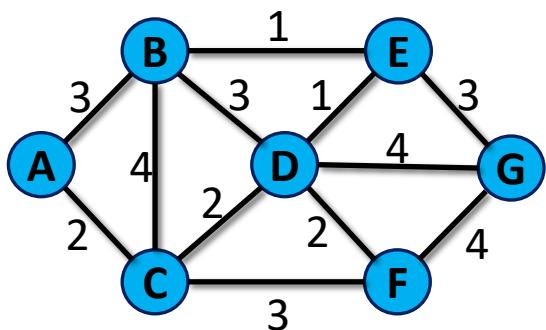
# Kruskal's MST Algorithm



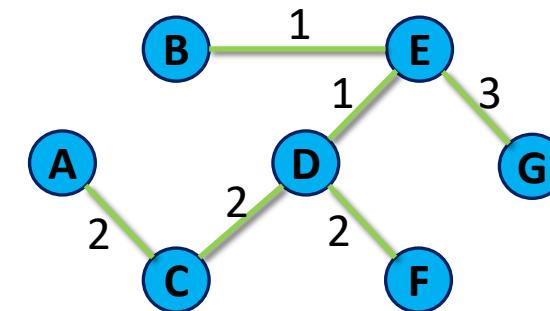
Edges	Cost
BE	1
DE	1
AC	2
CD	2
DF	2
AB	3
BD	3
CF	3
EG	3
BC	4
DG	4
FG	4



# Kruskal's MST Algorithm



Edges	Cost
BE	1
DE	1
AC	2
CD	2
DF	2
AB	3
BD	3
CF	3
EG	3
BC	4
DG	4
FG	4



Total Cost =  $1+1+2+2+3 = 11$

# Prim's MST Algorithm

---

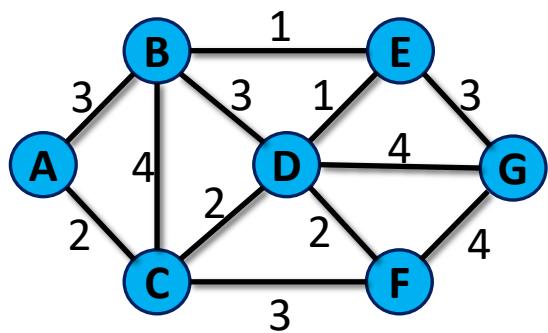
## Algorithm

1. Start with any vertex; add it to the MST.
2. While MST doesn't include all vertices:
  - a. Pick the smallest edge connecting MST to a new vertex.
  - b. Add that edge and vertex to the MST.
3. Repeat until all vertices are included.

# Prim's MST Algorithm

---

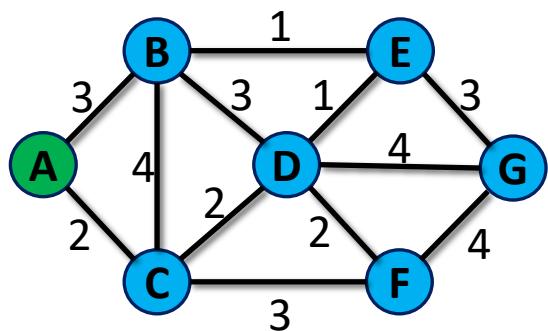
1. Start With Vertex A



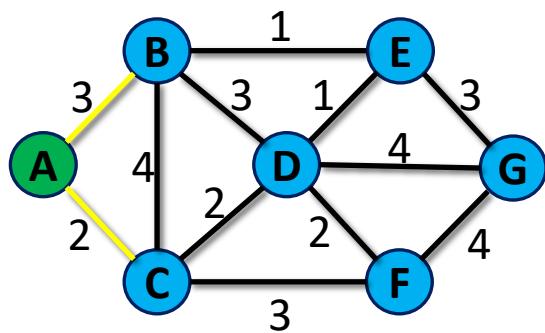
# Prim's MST Algorithm

1. Start With Vertex A

2. Explore the connected edges to MST



# Prim's MST Algorithm



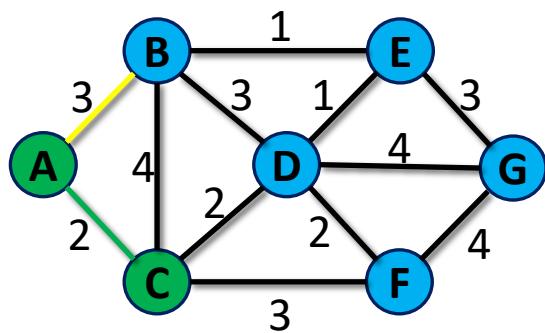
1. Start With Vertex A

2. Explore the connected edges to MST

3. Select minimum edge then add that vertex and edges to MST

AC - 2	AB - 3						
--------	--------	--	--	--	--	--	--

# Prim's MST Algorithm



1. Start With Vertex A

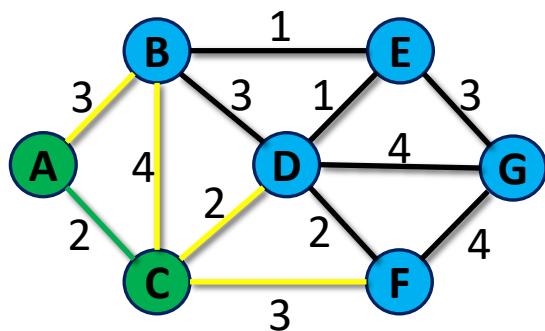
2. Explore the connected edges to MST

3. Select minimum edge then add that vertex and edges to MST

4. Repeat 2 and 3 until all vertices are included in MST.

AB - 3							
--------	--	--	--	--	--	--	--

# Prim's MST Algorithm



1. Start With Vertex A

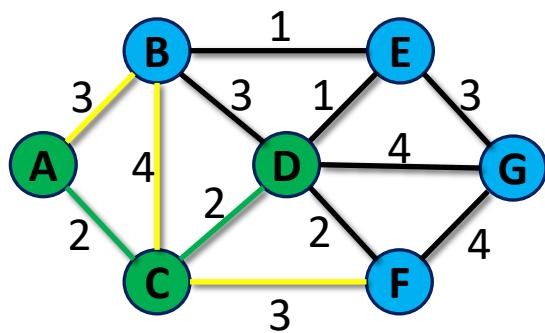
2. Explore the connected edges to MST

3. Select minimum edge then add that vertex and edges to MST

4. Repeat 2 and 3 until all vertices are included in MST.

CD - 2	AB - 3	CF - 3	CB - 4				
--------	--------	--------	--------	--	--	--	--

# Prim's MST Algorithm



1. Start With Vertex A

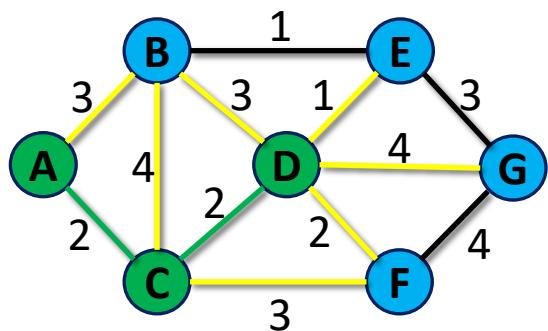
2. Explore the connected edges to MST

3. Select minimum edge then add that vertex and edges to MST

4. Repeat 2 and 3 until all vertices are included in MST.

AB - 3	CF - 3	CB - 4					
--------	--------	--------	--	--	--	--	--

# Prim's MST Algorithm



1. Start With Vertex A

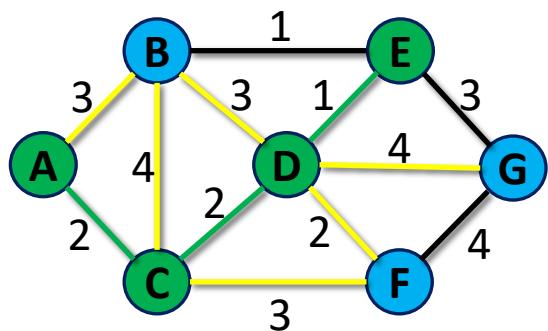
2. Explore the connected edges to MST

3. Select minimum edge then add that vertex and edges to MST

4. Repeat 2 and 3 until all vertices are included in MST.

DE - 1	DF - 2	AB - 3	CF - 3	DB - 3	CB - 4	DG - 4	
--------	--------	--------	--------	--------	--------	--------	--

# Prim's MST Algorithm



1. Start With Vertex A

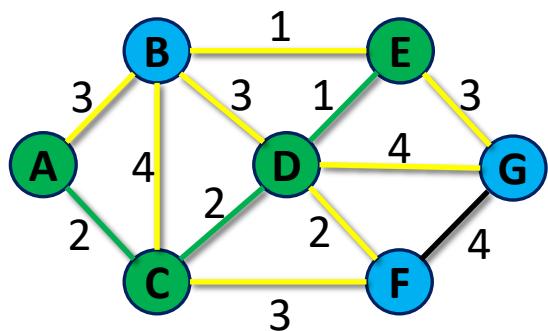
2. Explore the connected edges to MST

3. Select minimum edge then add that vertex and edges to MST

4. Repeat 2 and 3 until all vertices are included in MST.

DF - 2	AB - 3	CF - 3	DB - 3	CB - 4	DG - 4		
--------	--------	--------	--------	--------	--------	--	--

# Prim's MST Algorithm



1. Start With Vertex A

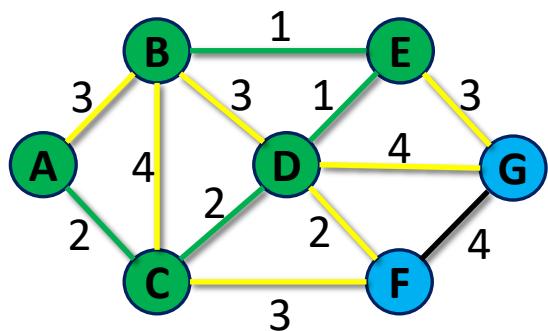
2. Explore the connected edges to MST

3. Select minimum edge then add that vertex and edges to MST

4. Repeat 2 and 3 until all vertices are included in MST.

EB - 1	DF - 2	AB - 3	CF - 3	DB - 3	EG - 3	CB - 4	DG - 4
--------	--------	--------	--------	--------	--------	--------	--------

# Prim's MST Algorithm



1. Start With Vertex A

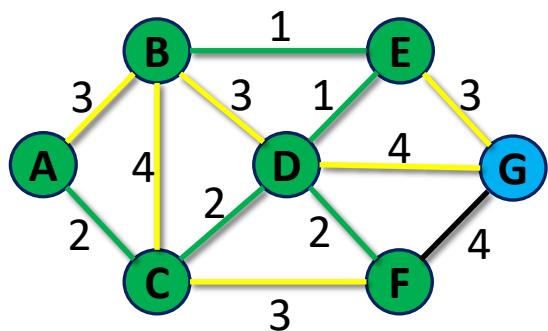
2. Explore the connected edges to MST

3. Select minimum edge then add that vertex and edges to MST

4. Repeat 2 and 3 until all vertices are included in MST.

DF - 2	AB - 3	CF - 3	DB - 3	EG - 3	CB - 4	DG - 4	
--------	--------	--------	--------	--------	--------	--------	--

# Prim's MST Algorithm



1. Start With Vertex A

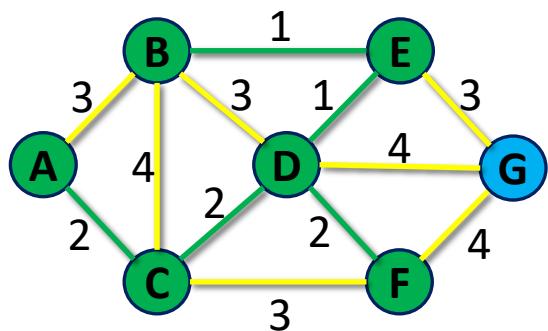
2. Explore the connected edges to MST

3. Select minimum edge then add that vertex and edges to MST

4. Repeat 2 and 3 until all vertices are included in MST.

AB - 3	CF - 3	DB - 3	EG - 3	CB - 4	DG - 4		
--------	--------	--------	--------	--------	--------	--	--

# Prim's MST Algorithm



1. Start With Vertex A

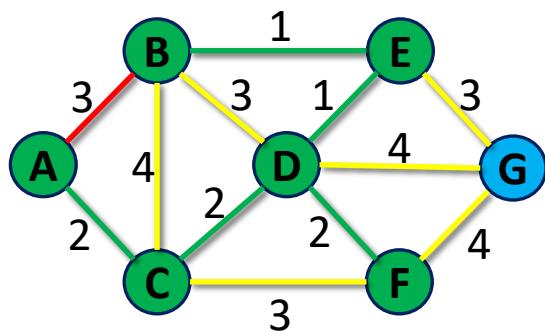
2. Explore the connected edges to MST

3. Select minimum edge then add that vertex and edges to MST

4. Repeat 2 and 3 until all vertices are included in MST.

AB - 3	CF - 3	DB - 3	EG - 3	CB - 4	DG - 4	FG - 4	
--------	--------	--------	--------	--------	--------	--------	--

# Prim's MST Algorithm



1. Start With Vertex A

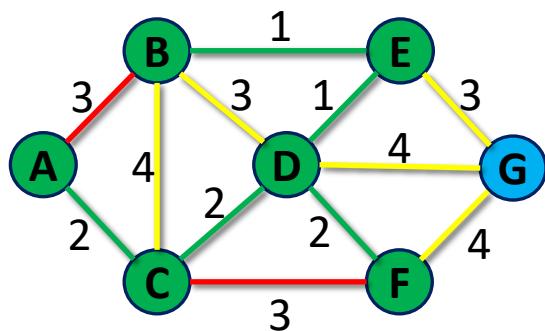
2. Explore the connected edges to MST

3. Select minimum edge then add that vertex and edges to MST

4. Repeat 2 and 3 until all vertices are included in MST.

CF - 3	DB - 3	EG - 3	CB - 4	DG - 4	FG - 4		
--------	--------	--------	--------	--------	--------	--	--

# Prim's MST Algorithm



1. Start With Vertex A

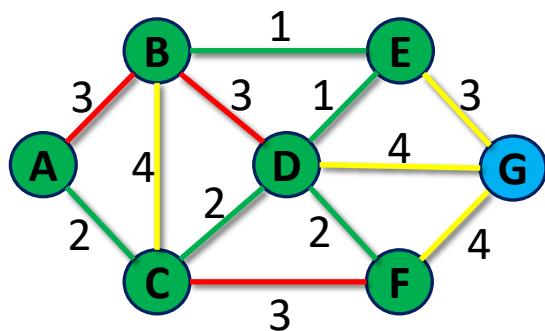
2. Explore the connected edges to MST

3. Select minimum edge then add that vertex and edges to MST

4. Repeat 2 and 3 until all vertices are included in MST.

DB - 3	EG - 3	CB - 4	DG - 4	FG - 4			
--------	--------	--------	--------	--------	--	--	--

# Prim's MST Algorithm



1. Start With Vertex A

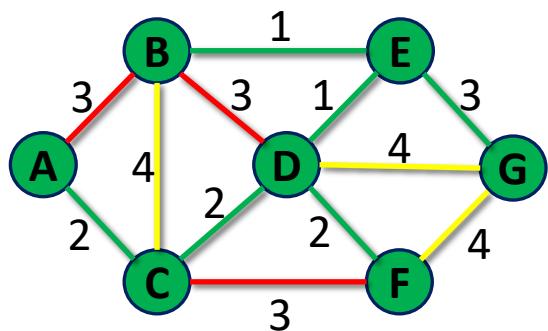
2. Explore the connected edges to MST

3. Select minimum edge then add that vertex and edges to MST

4. Repeat 2 and 3 until all vertices are included in MST.

EG - 3	CB - 4	DG - 4	FG - 4				
--------	--------	--------	--------	--	--	--	--

# Prim's MST Algorithm



1. Start With Vertex A

2. Explore the connected edges to MST

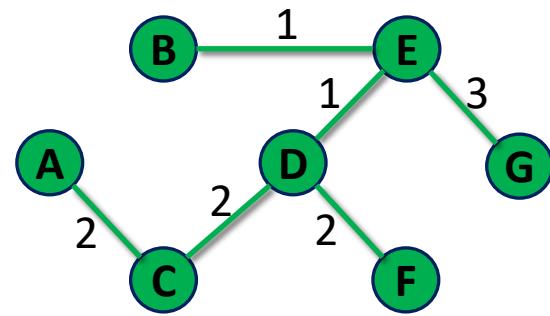
3. Select minimum edge then add that vertex and edges to MST

4. Repeat 2 and 3 until all vertices are included in MST.

CB - 4	DG - 4	FG - 4					
--------	--------	--------	--	--	--	--	--

# Prim's MST Algorithm

---



So, the cost of the MST is =  $2+2+2+1+1+3 = \mathbf{11}$

*THANK YOU*

---