CSCI 232:
Data Structures and Algorithms

Minimum Spanning Tree (MST) Part 1

Reese Pearsall

Spring 2024

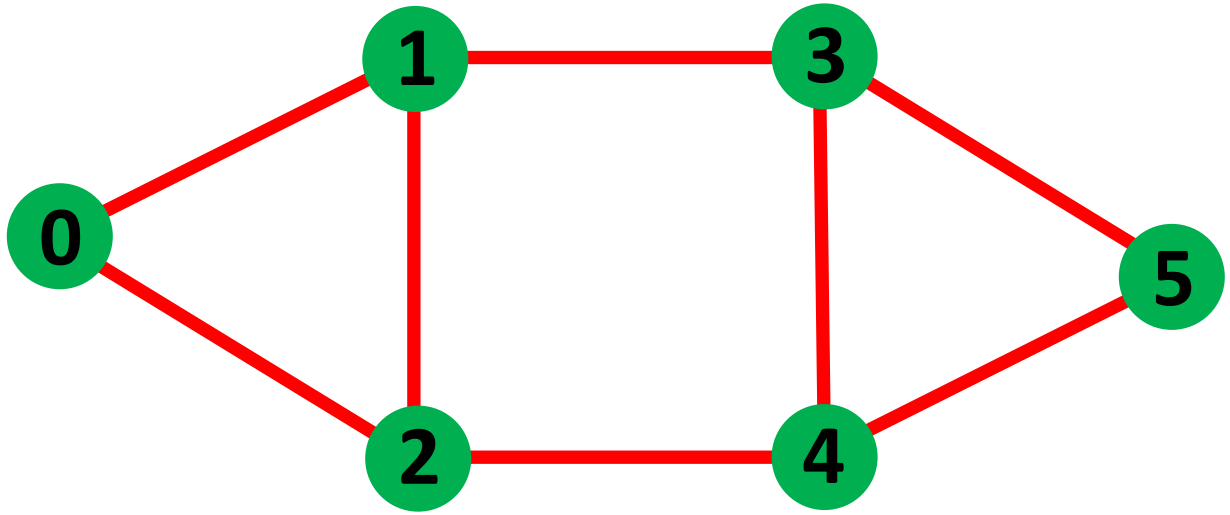
Announcements

Lab 9 due on Friday

Fill out survey if you haven’t already for lab 8

Graphs

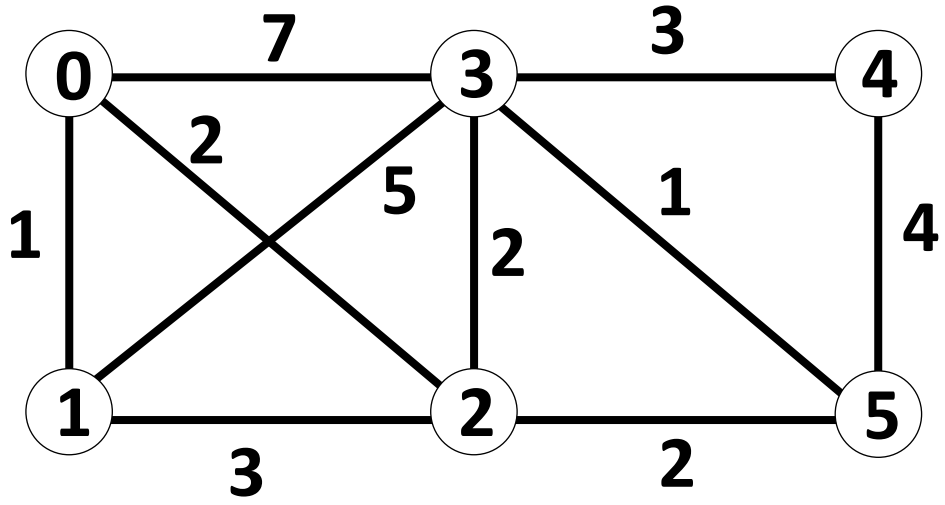
G = (V, E)



Adjacency List

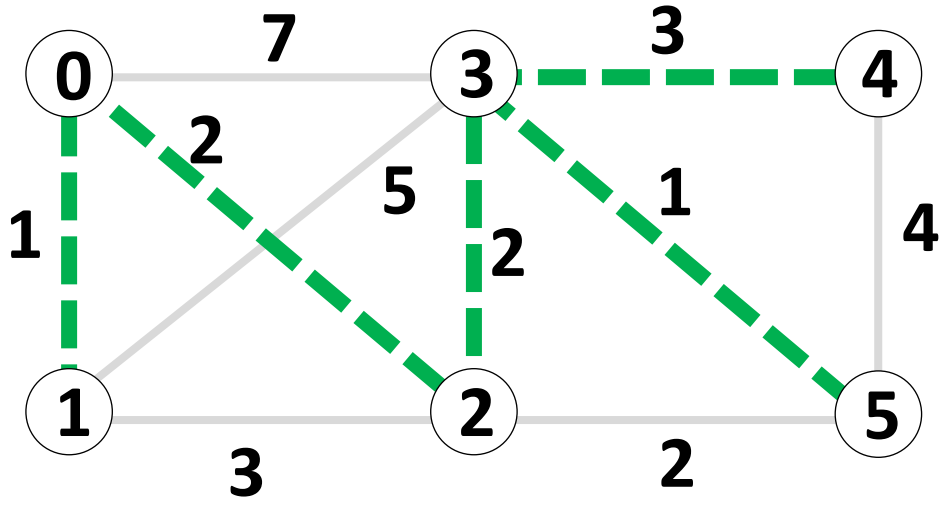
0	→	{1,2}
1	→	{0,2,3}
2	→	{0,1,4}
3	→	{1,4,5}
4	→	{2,3,5}
5	→	{3,4}

Minimum Spanning Tree



Edge-weighted graph: A graph where each edge has a weight (cost).

Minimum Spanning Tree

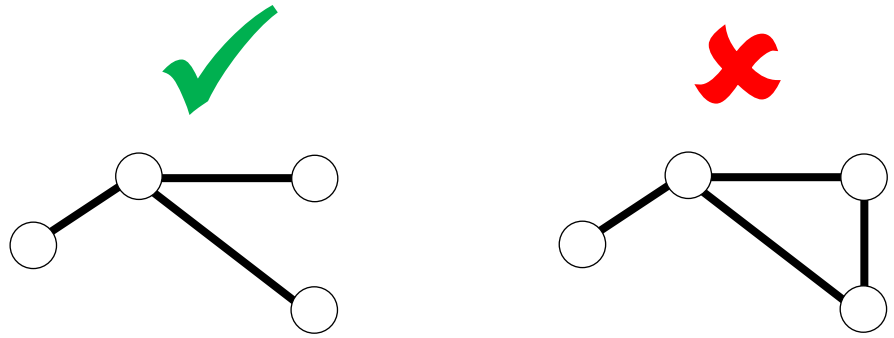


Edge-weighted graph: A graph where each edge has a weight (cost).

MST Goal: Connect all vertices to each other with a minimum weight subset of edges.

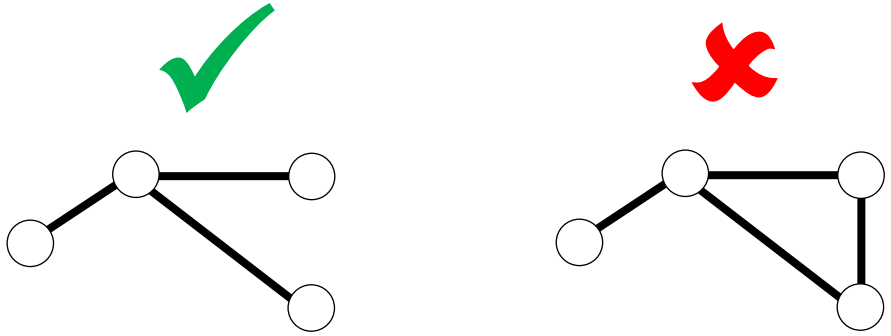
Minimum Spanning Tree

Tree – connected graph with no loops.



Minimum Spanning Tree

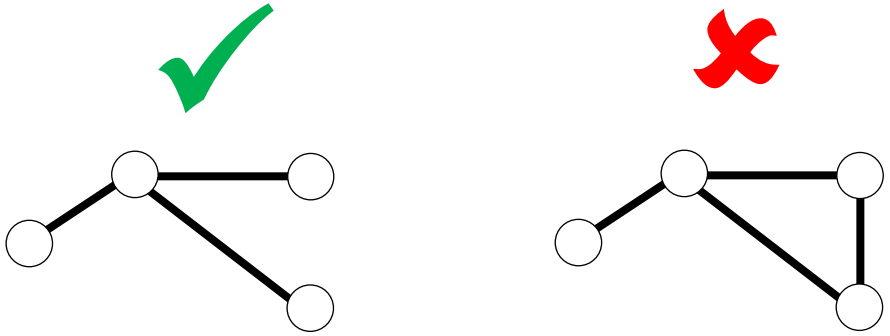
Tree – connected graph with no loops.



Spanning tree – tree that includes all vertices in a graph.

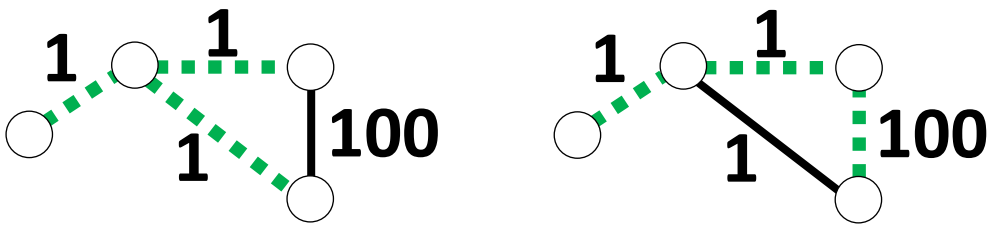
Minimum Spanning Tree

Tree – connected graph with no loops.

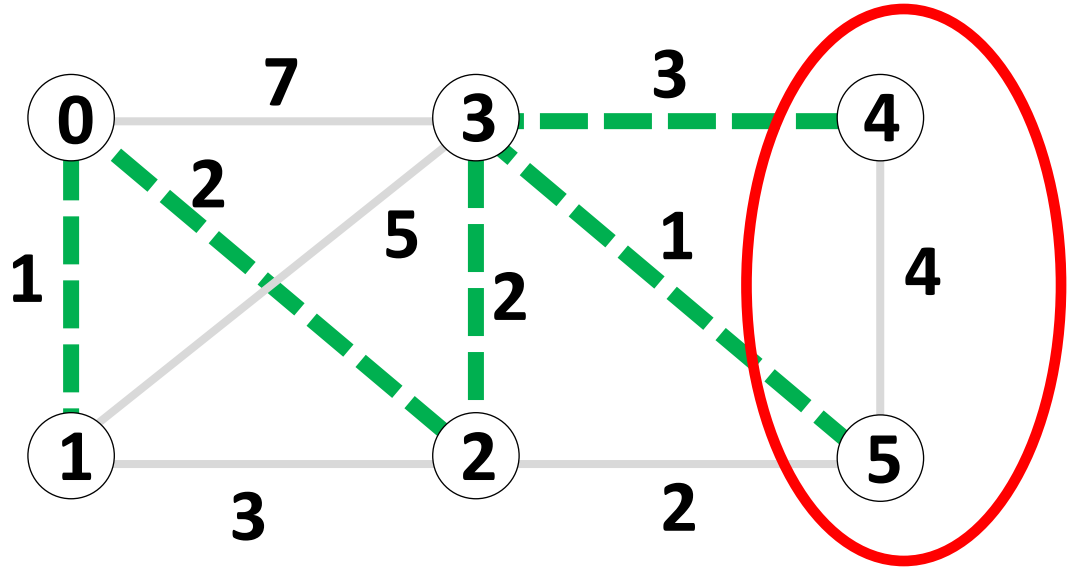


Spanning tree – tree that includes all vertices in a graph.

Minimum spanning tree – spanning tree whose sum of edge costs is the minimum possible value.



Minimum Spanning Tree



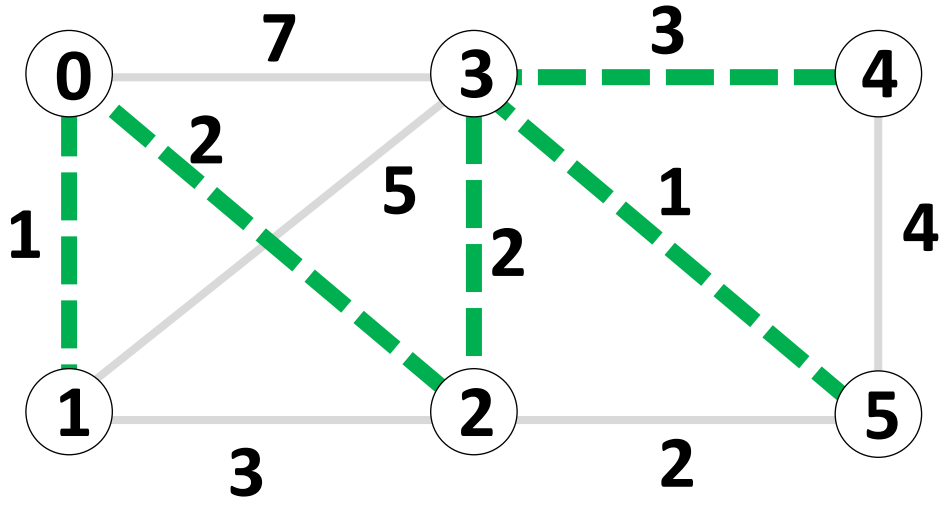
Does it ever make sense to have a cycle?

No!

Must be a tree!

MST Goal: Connect all vertices to each other with a minimum weight subset of edges.

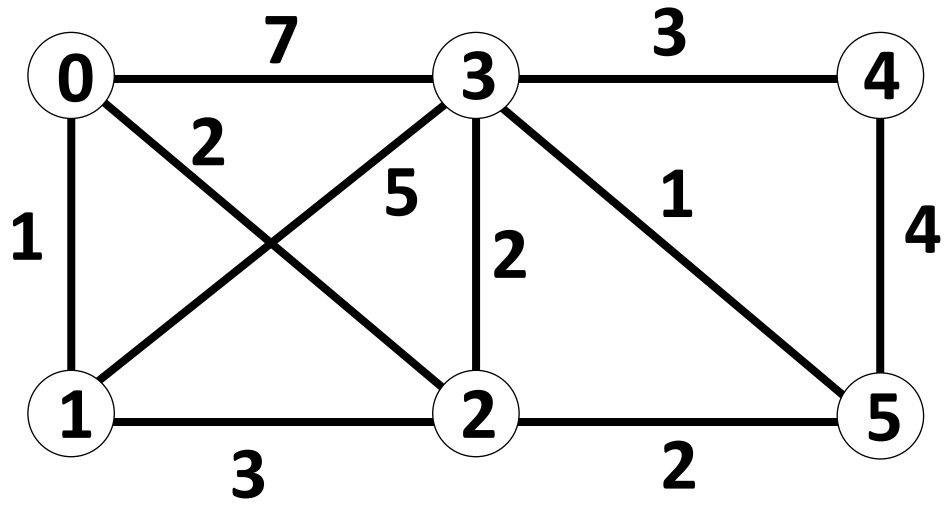
Minimum Spanning Tree



How to find MSTs?

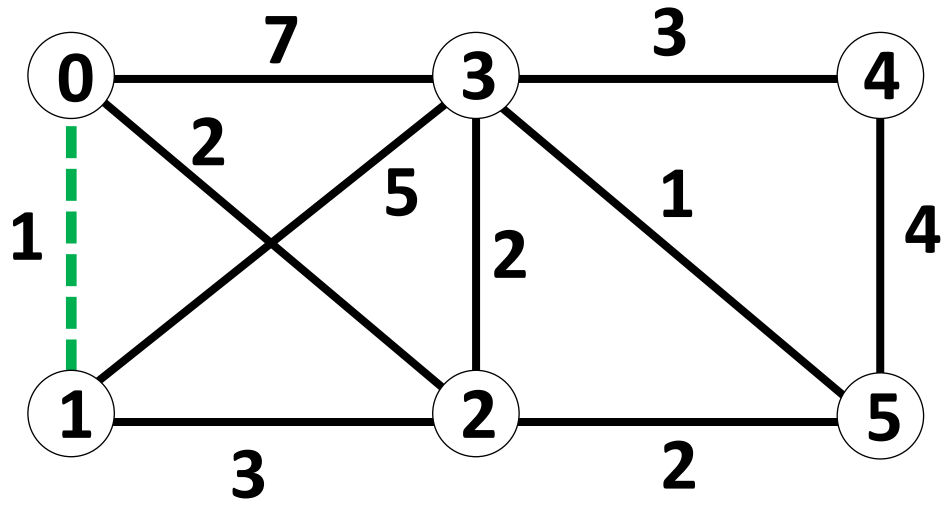
Kruskal's MST Algorithm

At each iteration, add the edge with smallest weight, that does not create a cycle.



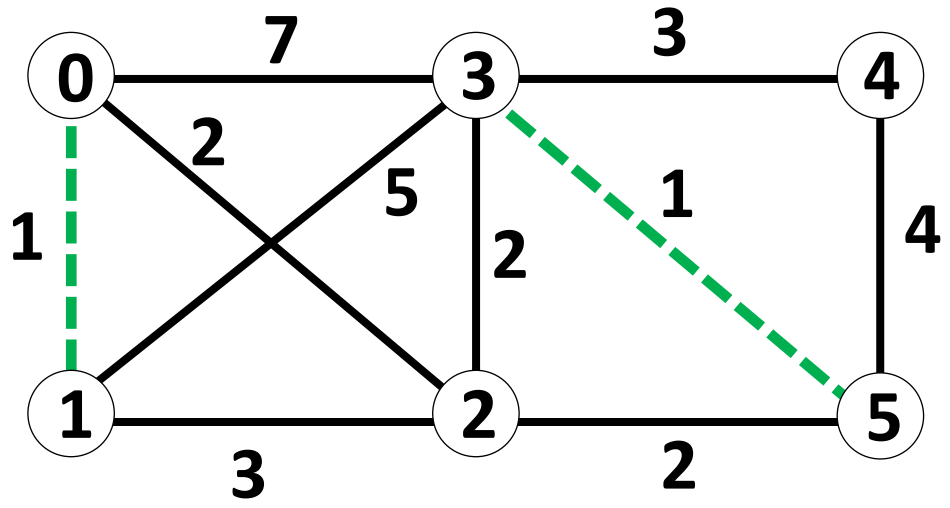
Kruskal's MST Algorithm

At each iteration, add the edge with smallest weight, that does not create a cycle.



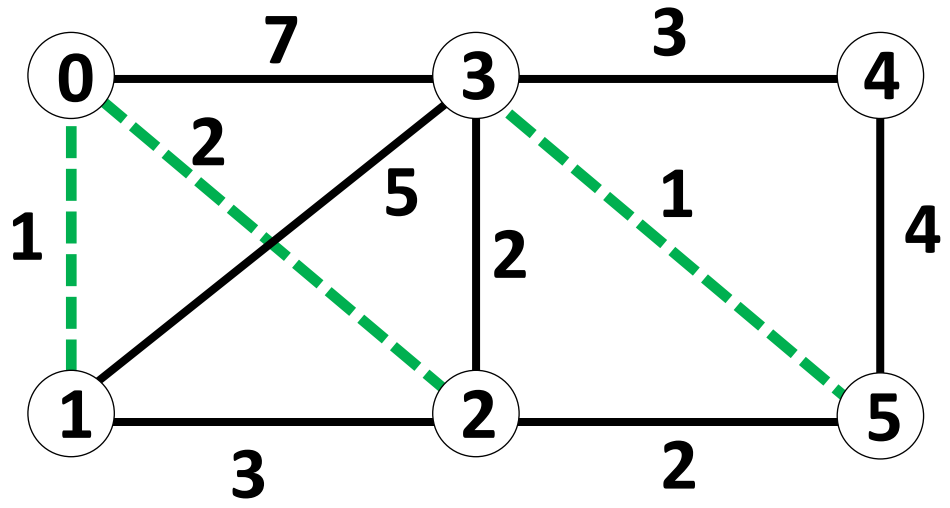
Kruskal's MST Algorithm

At each iteration, add the edge with smallest weight, that does not create a cycle.



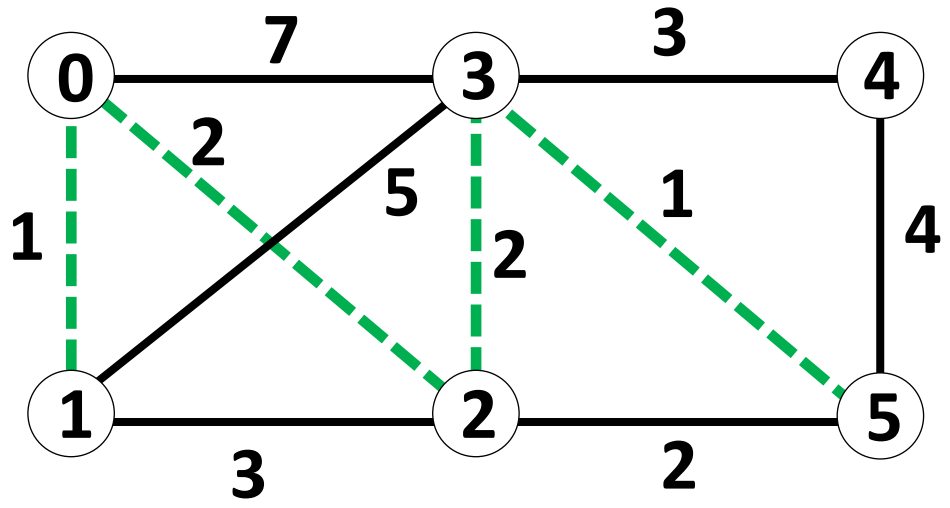
Kruskal's MST Algorithm

At each iteration, add the edge with smallest weight, that does not create a cycle.



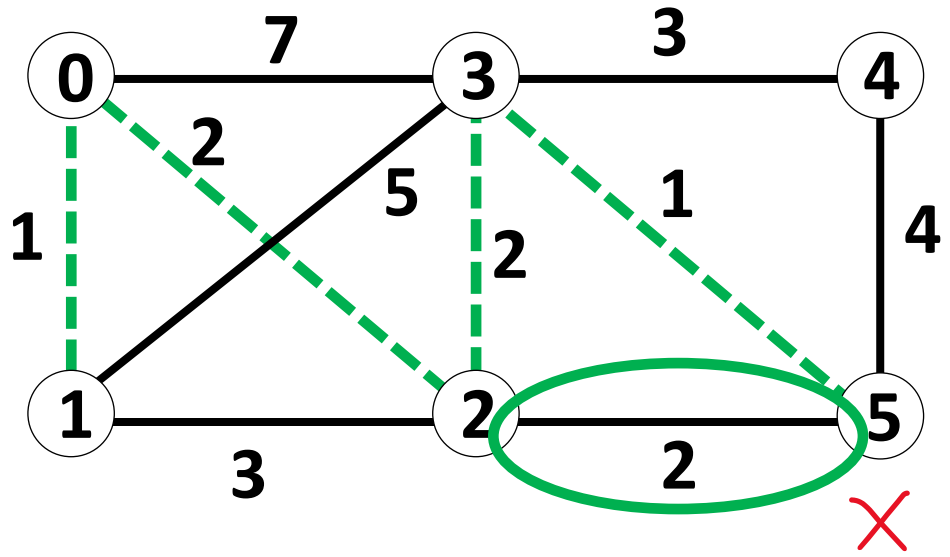
Kruskal's MST Algorithm

At each iteration, add the edge with smallest weight, that does not create a cycle.



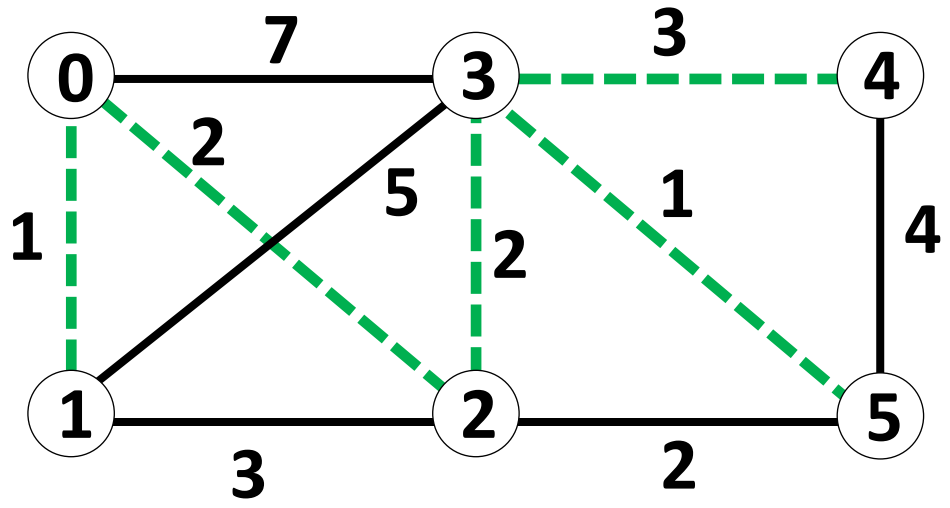
Kruskal's MST Algorithm

At each iteration, add the edge with smallest weight, that does not create a cycle.



Kruskal's MST Algorithm

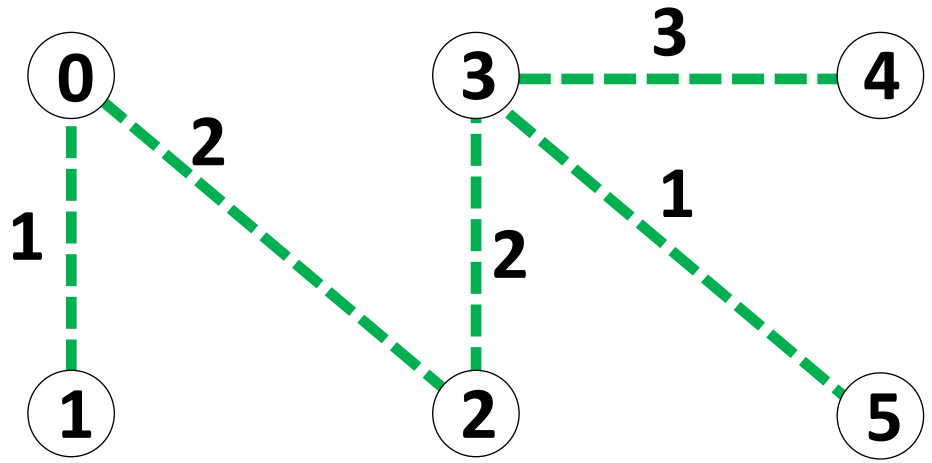
At each iteration, add the edge with smallest weight, that does not create a cycle.



Kruskal's MST Algorithm

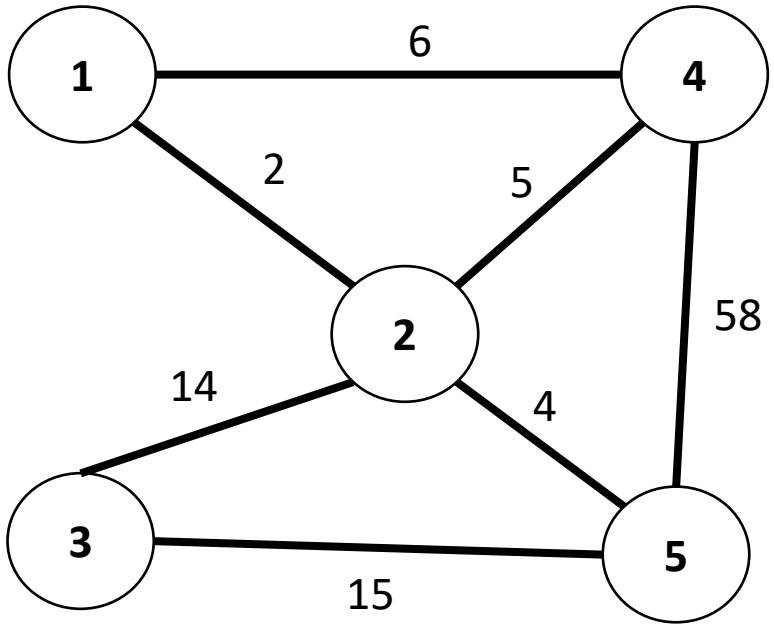
At each iteration, add the edge with smallest weight, that does not create a cycle.

MST = [0, 1], [0, 2], [2,3], [3,5], [3,4] **Total Cost = 9**



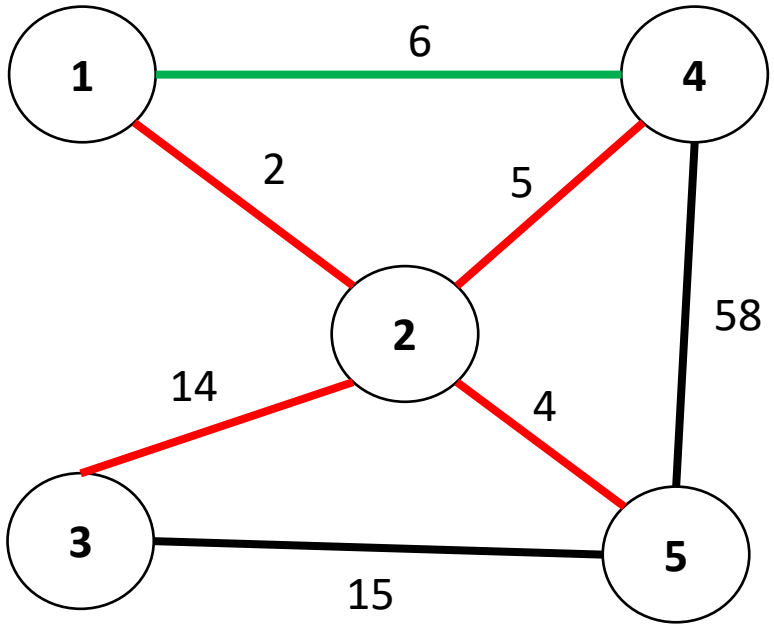
MST vs Shortest Path

MST and shortest path are two different problems, and sometimes that shortest path will not be part of the MST



MST vs Shortest Path

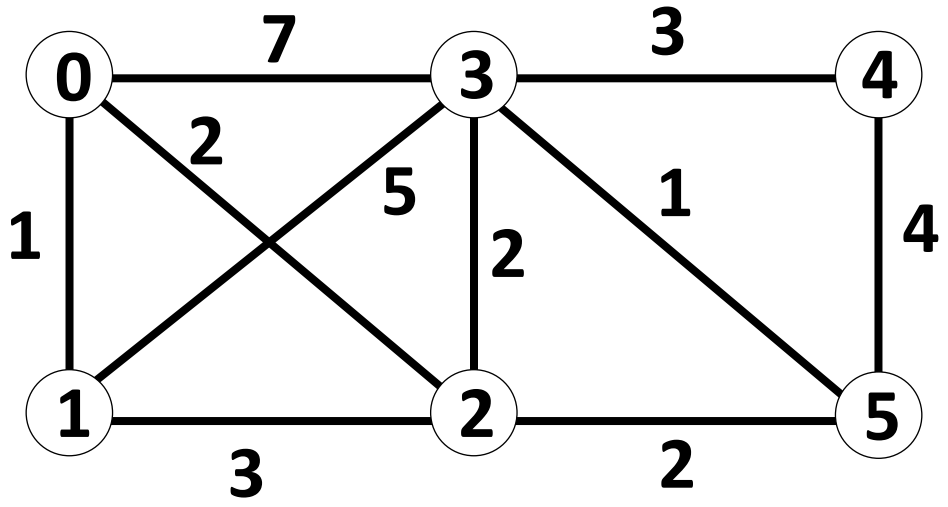
MST and shortest path are two different problems, and sometimes that shortest path will not be part of the MST



MST Cost = 25

Shortest Path from 1 to 4 = 6

Weighted Graph




```
public class Edge {

        private int vertex1;
        private int vertex2;

        private int weight;

        public int[] getVertices()

        public int getWeight()

        public String toString()

        public boolean equals()
```