EMTH403

Mathematical Foundation for Computer Science

Nitin K. Mishra (Ph.D.)

Associate Professor

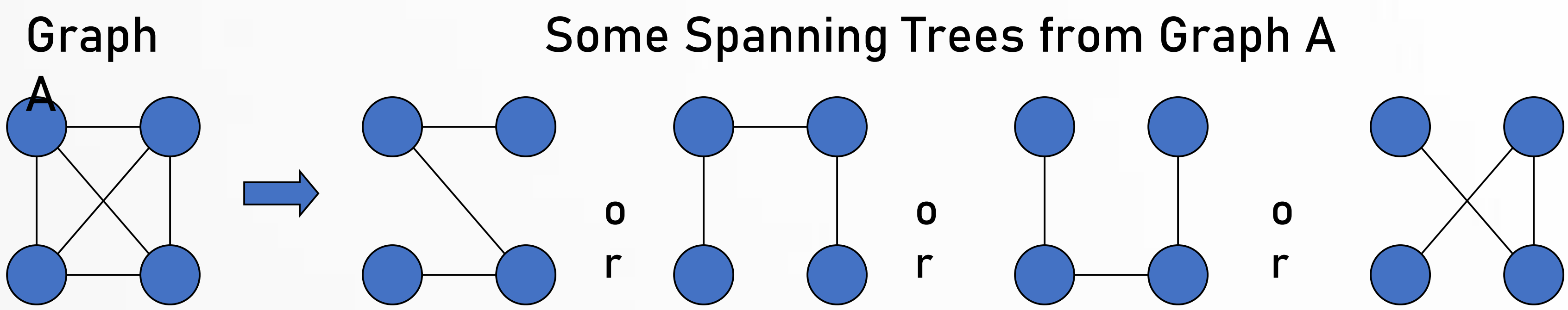
Lecture Outcomes

After this lecture, you will be able to

- understand what Spanning Trees
- understand what is Kruskal's Algorithm.

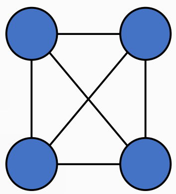
Spanning Trees

- A spanning tree of a graph is just a subgraph that contains all the vertices and is a tree.
- A graph may have many spanning trees.

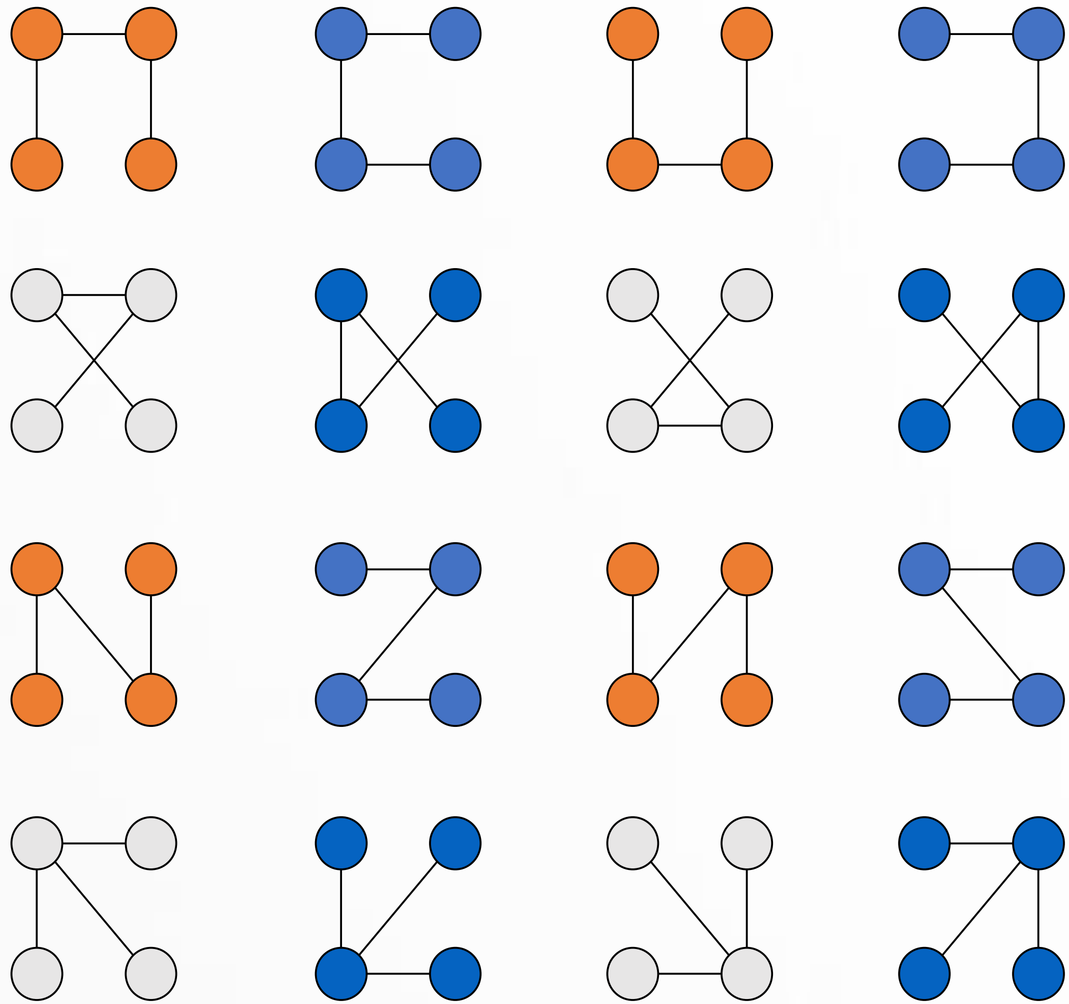


Spanning Trees

Complete Graph



All 16 of its Spanning Trees

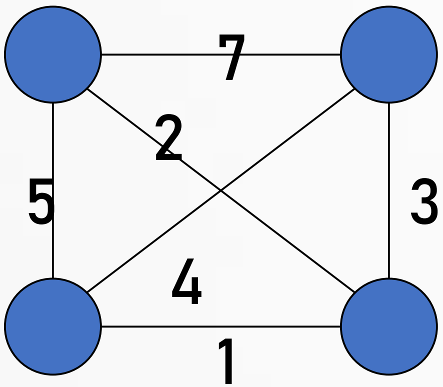


Content Source: **Discrete Mathematics** and **Its Applications**. Seventh Edition. **Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University.** (and formerly AT&T Laboratories)

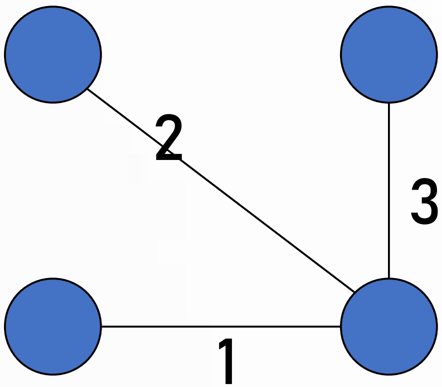
Minimum Spanning Trees

The Minimum Spanning Tree for a given graph is the Spanning Tree of minimum cost for that graph.

Complete Graph

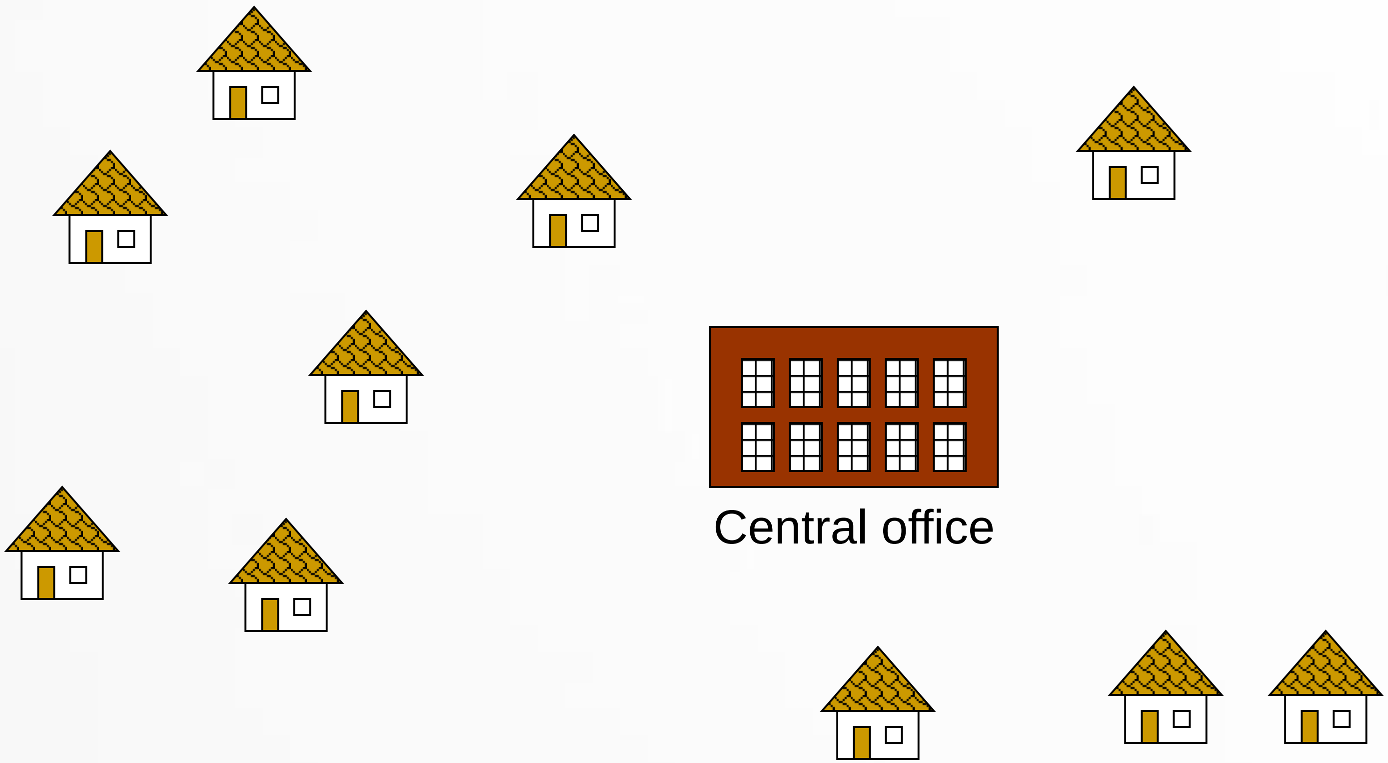


Minimum Spanning Tree

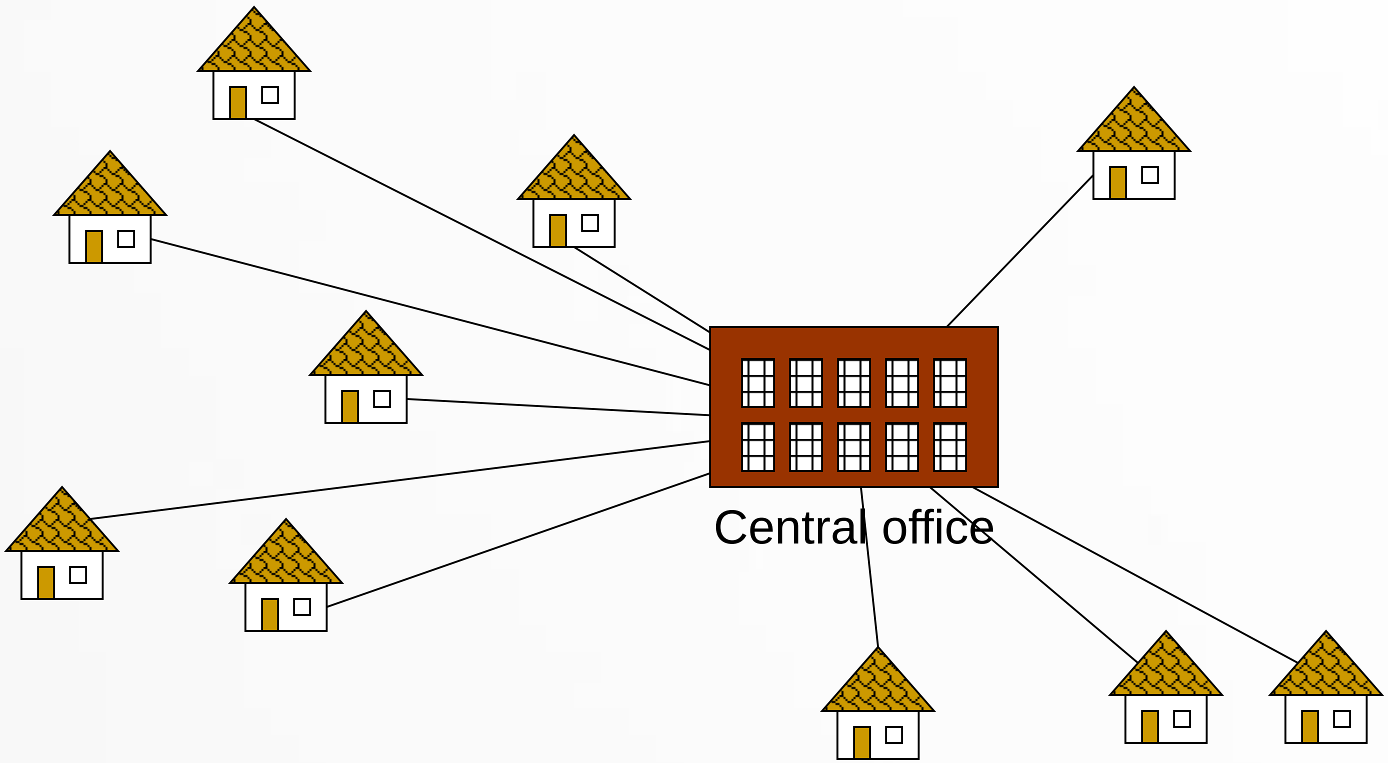


Content Source: Discrete Mathematics and Its Applications. Seventh Edition. Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University. (and formerly AT&T Laboratories)

Spanning Trees



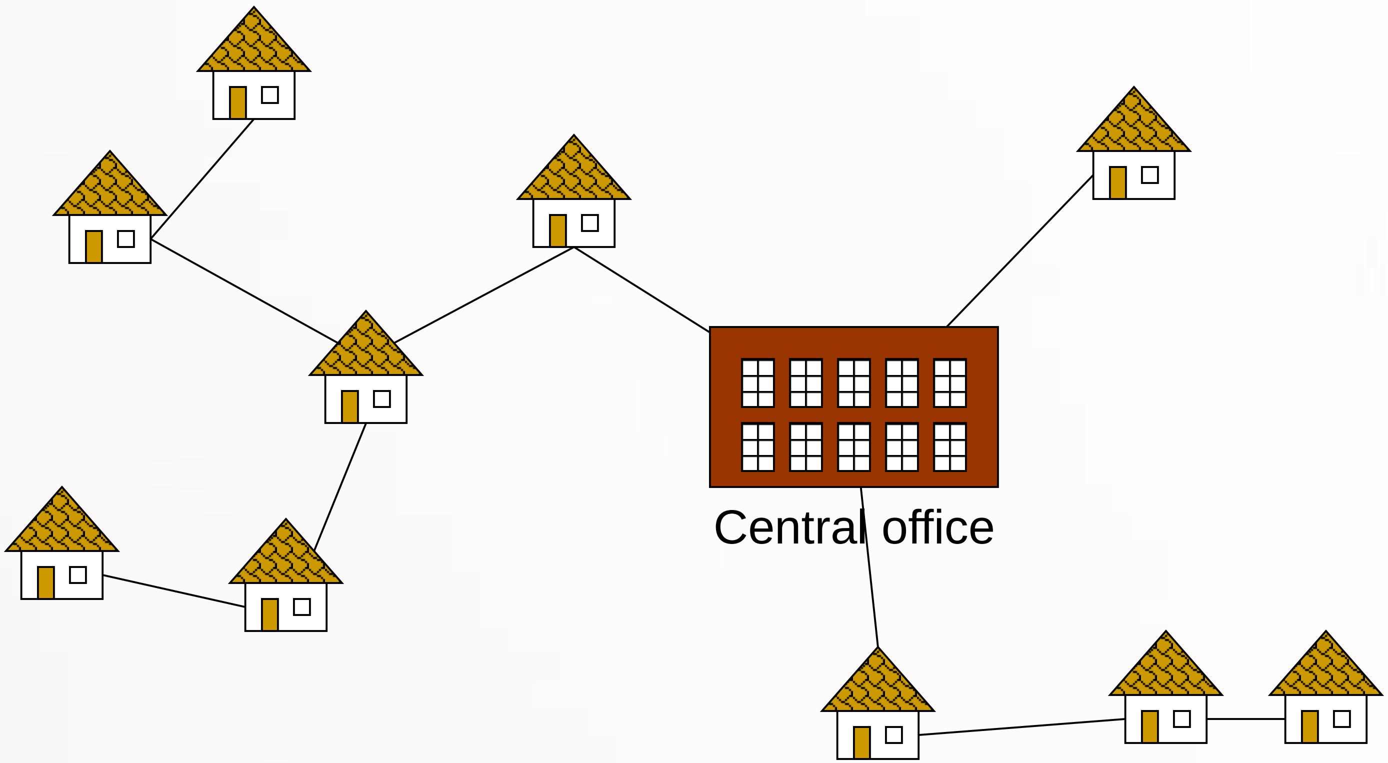
Spanning Trees



Expensive!

Content Source: **Discrete Mathematics** and **Its Applications**. Seventh Edition. **Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University.** (and formerly AT&T Laboratories)

Spanning Trees



Minimize the total length of wire connecting the customers

Algorithms for Obtaining the Minimum Spanning Tree

Kruskal's Algorithm

Prim's Algorithm

Spanning Trees

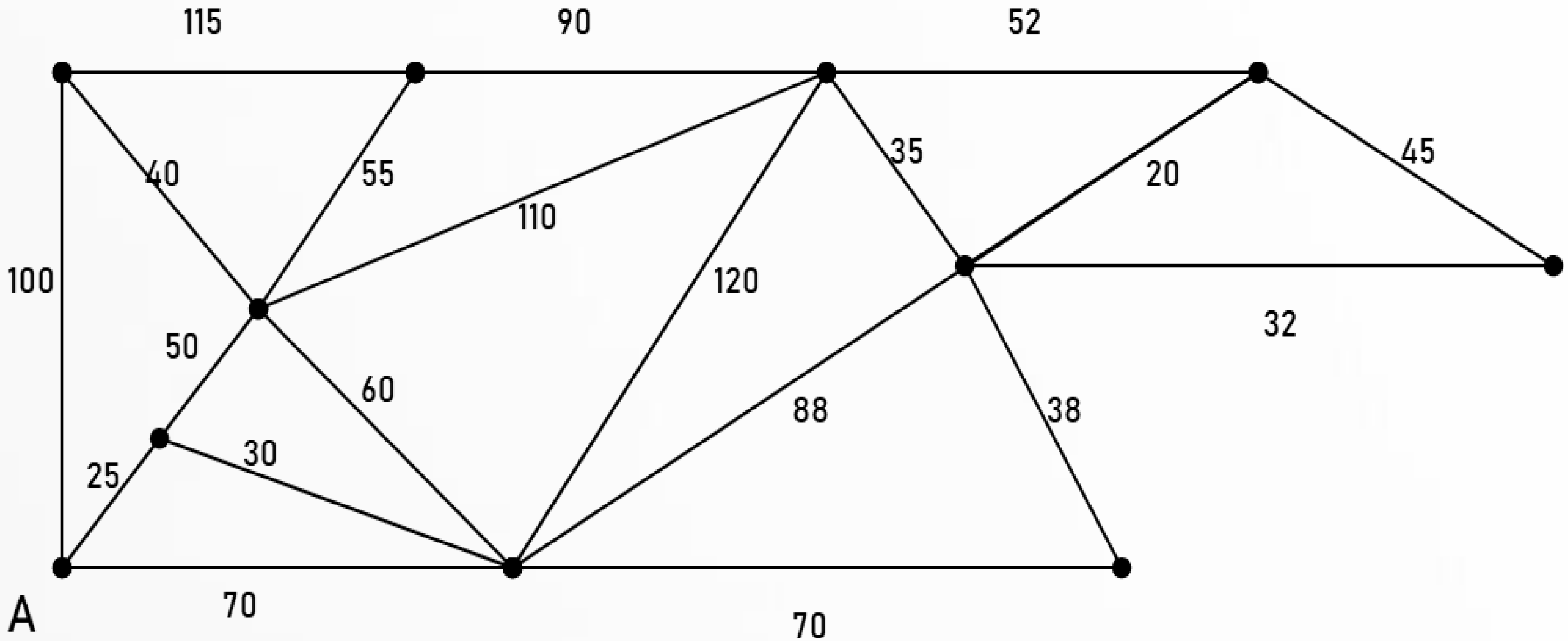
- There are two basic algorithms for finding minimum-cost spanning trees, and both are greedy algorithms
- Kruskal's algorithm: Start with no nodes or edges in the spanning tree, and repeatedly add the cheapest edge that does not create a cycle
    - Here, we consider the spanning tree to consist of edges only

Spanning Trees

- 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.
  - Here, we consider the spanning tree to consist of both nodes and edges

Kruskal's Algorithm

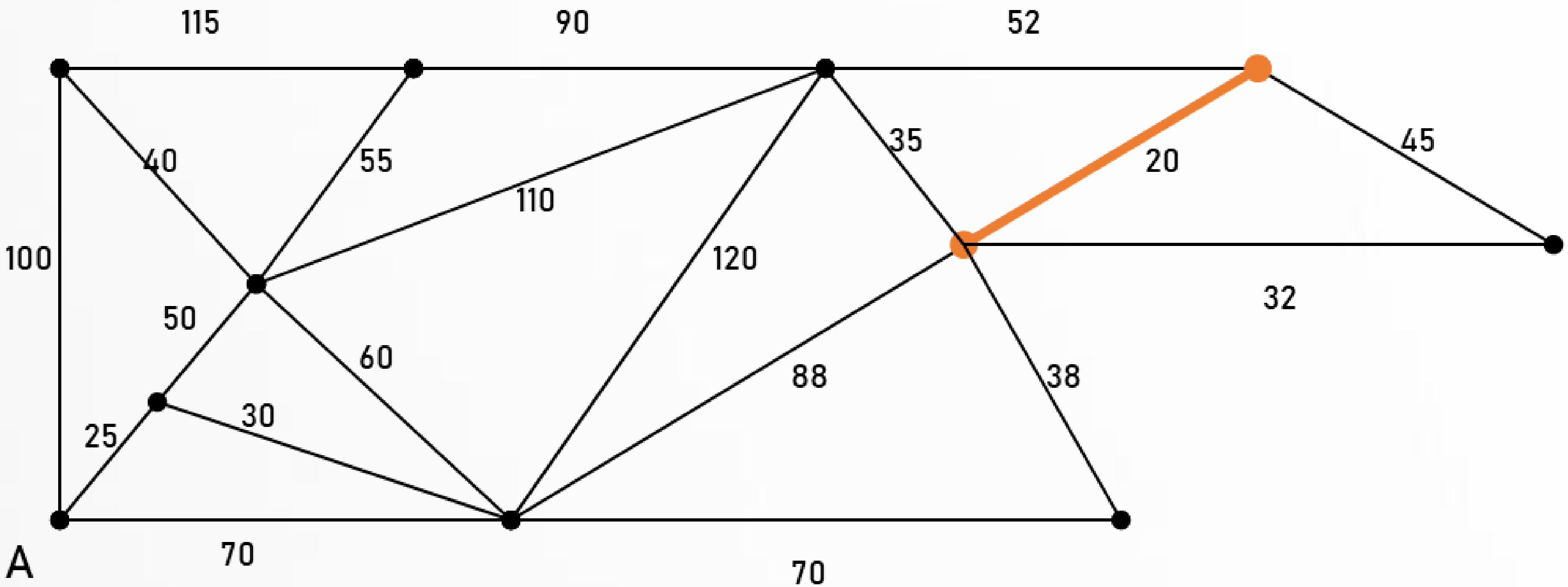
Find the minimum spanning tree using Kruskal's Algorithm.



List the edges in increasing order:

20, 25, 30, 32, 35, 38, 40, 45, 50, 52, 55, 60, 70, 70, 88, 90, 100, 110, 115, 120

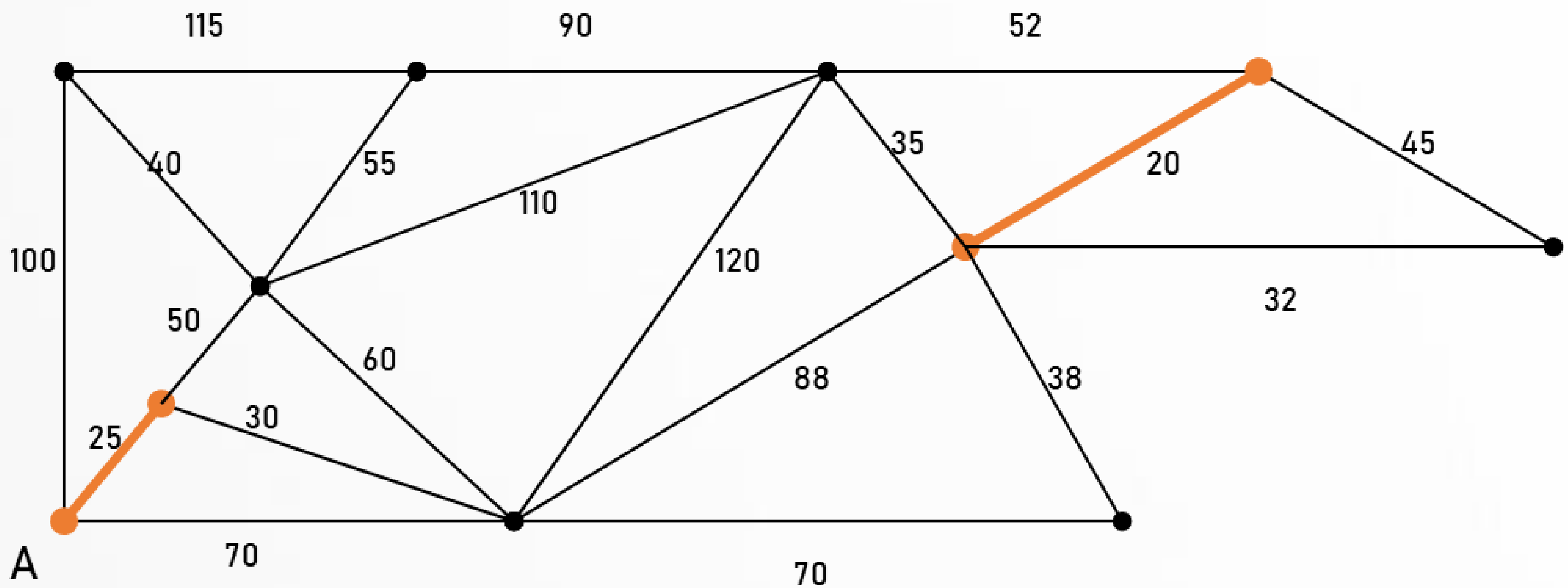
Spanning Trees



Starting from the left, add the edge to the tree if it does not close up a circuit with the edges chosen up to that point: 20, 25, 30, 32, 35, 38, 40, 45, 50, 52, 55, 60, 70, 70, 88, 90, 100, 110, 115, 120

Content Source: Discrete Mathematics and Its Applications. Seventh Edition. Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University. (and formerly AT&T Laboratories)

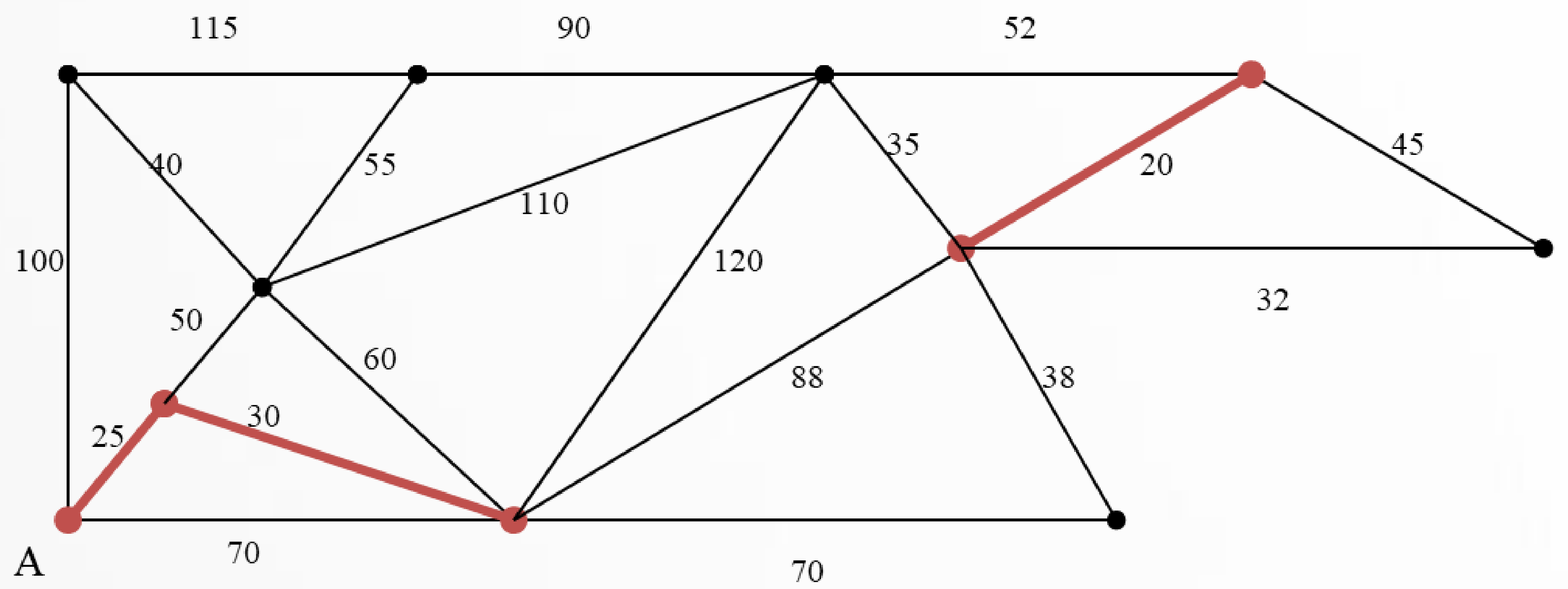
Kruskal's Algorithm



Add the next edge in the list to the tree if it does not close up a circuit with the edges chosen up to that point: 20, 25, 30, 32, 35, 38, 40, 45, 50, 52, 55, 60, 70, 70, 88, 90, 100, 110, 115, 120

Content Source: Discrete Mathematics and Its Applications. Seventh Edition. Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University. (and formerly AT&T Laboratories)

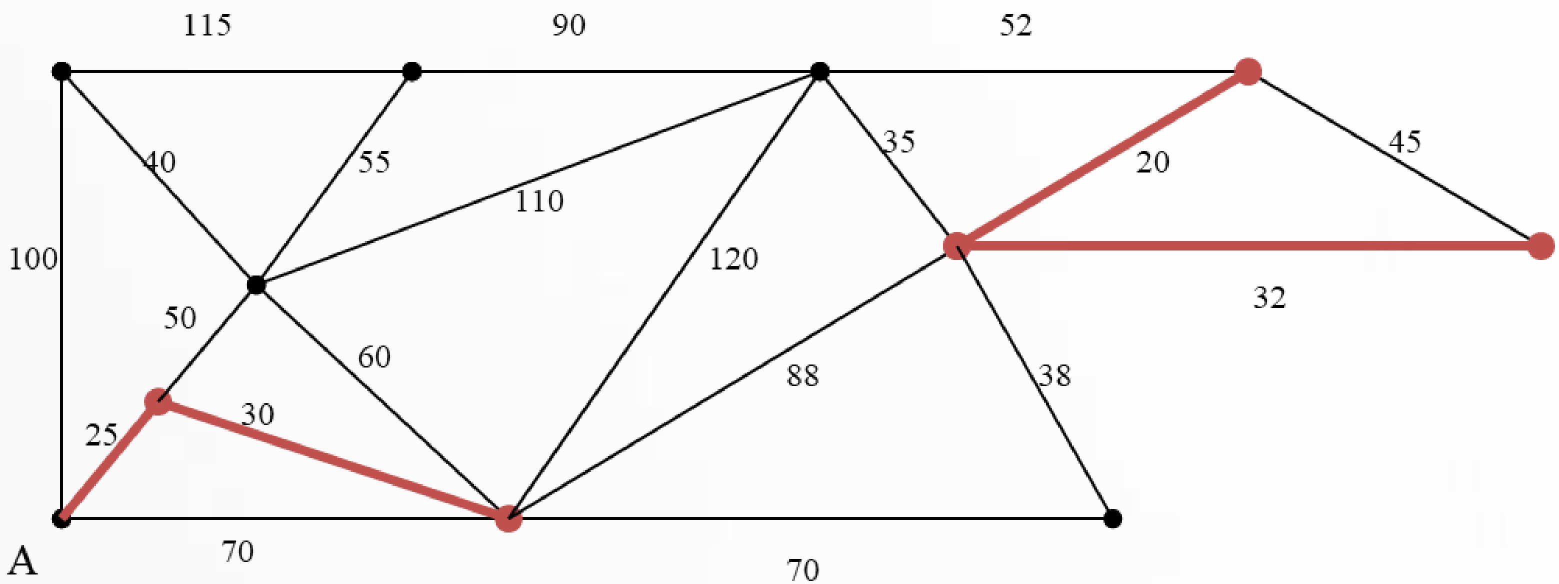
Kruskal's Algorithm



Add the next edge in the list to the tree if it does not close up a circuit with the edges chosen up to that point: 20, 25, 30, 32, 35, 38, 40, 45, 50, 52, 55, 60, 70, 70, 88, 90, 100, 110, 115, 120

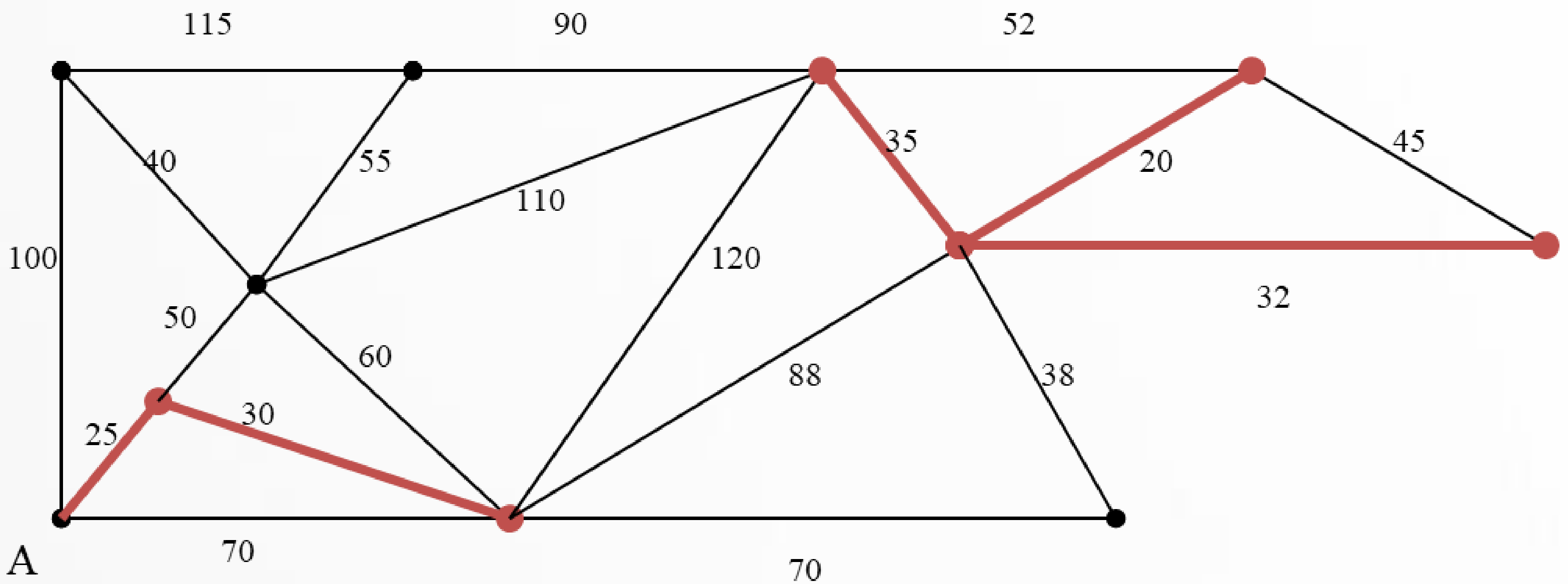
Content Source: **Discrete Mathematics** and **Its Applications**. Seventh Edition. **Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University.** (and formerly AT&T Laboratories)

Kruskal's Algorithm



Add the next edge in the list to the tree if it does not close up a circuit with the edges chosen up to that point: 20, 25, 30, 32, 35, 38, 40, 45, 50, 52, 55, 60, 70, 70, 88, 90, 100, 110, 115, 120

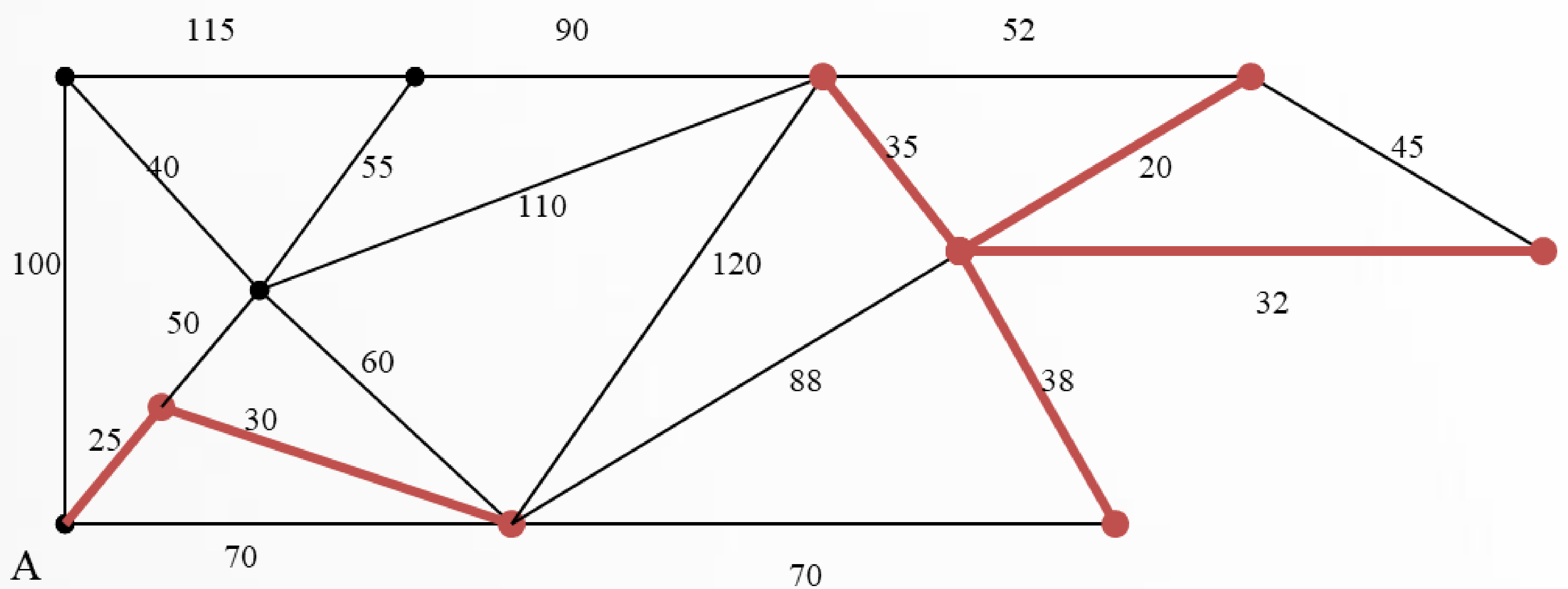
Kruskal's Algorithm



Add the next edge in the list to the tree if it does not close up a circuit with the edges chosen up to that point: 20, 25, 30, 32, 35, 38, 40, 45, 50, 52, 55, 60, 70, 70, 88, 90, 100, 110, 115, 120

Content Source: **Discrete Mathematics** and **Its Applications**. Seventh Edition. **Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University.** (and formerly AT&T Laboratories)

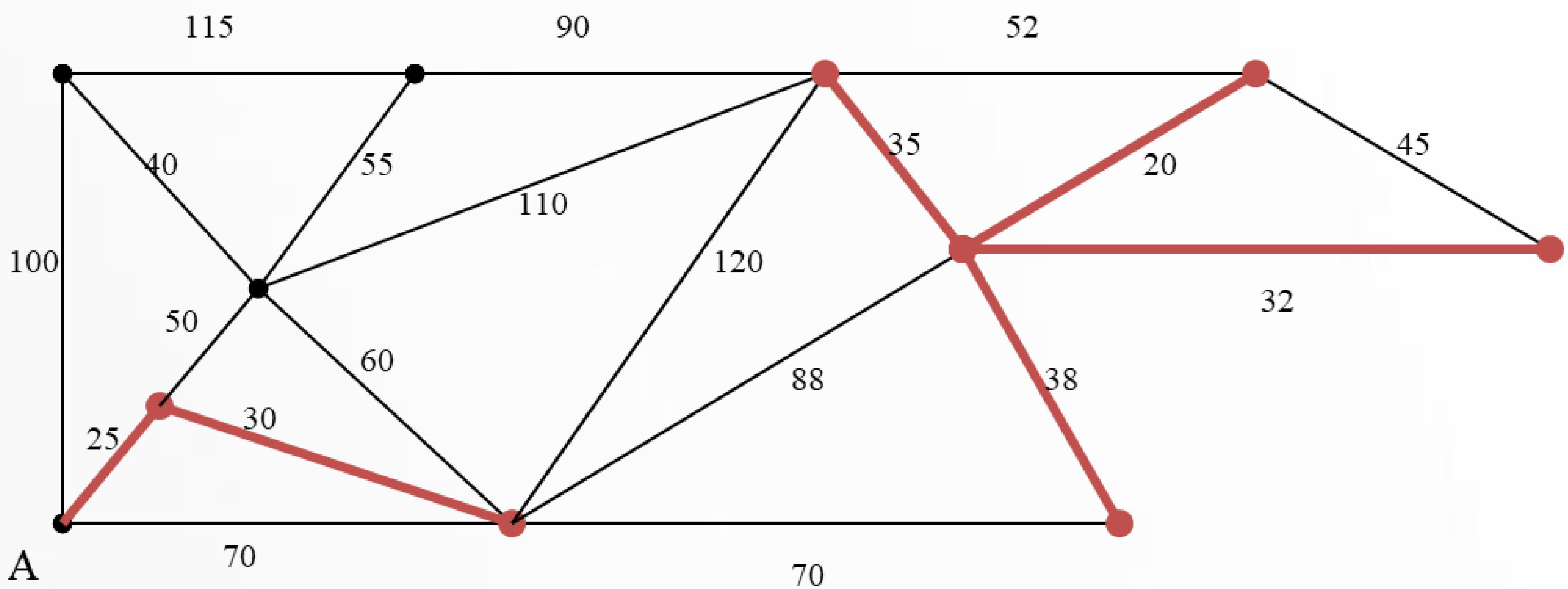
Kruskal's Algorithm



Add the next edge in the list to the tree if it does not close up a circuit with the edges chosen up to that point: 20, 25, 30, 32, 35, 38, 40, 45, 50, 52, 55, 60, 70, 70, 88, 90, 100, 110, 115, 120

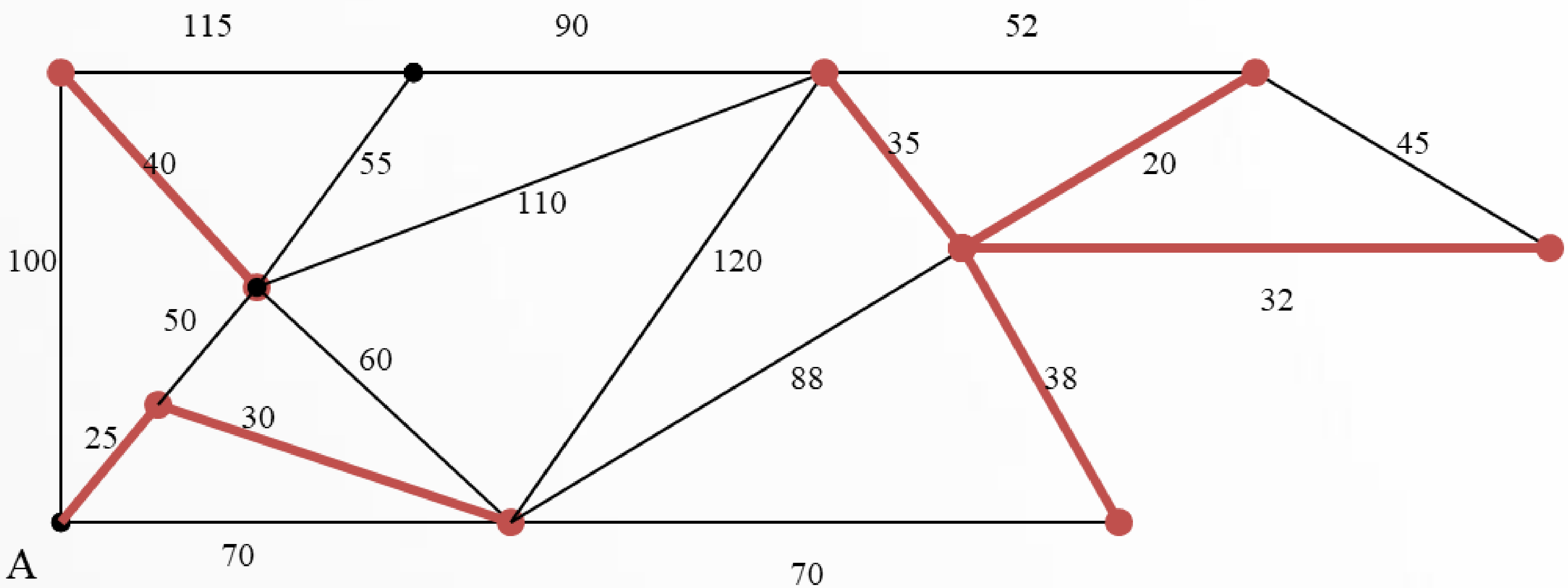
Content Source: **Discrete Mathematics** and **Its Applications**. Seventh Edition. **Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University.** (and formerly AT&T Laboratories)

Kruskal's Algorithm



Add the next edge in the list to the tree if it does not close up a circuit with the edges chosen up to that point: 20, 25, 30, 32, 35, 38, 40, 45, 50, 52, 55, 60, 70, 70, 88, 90, 100, 110, 115, 120

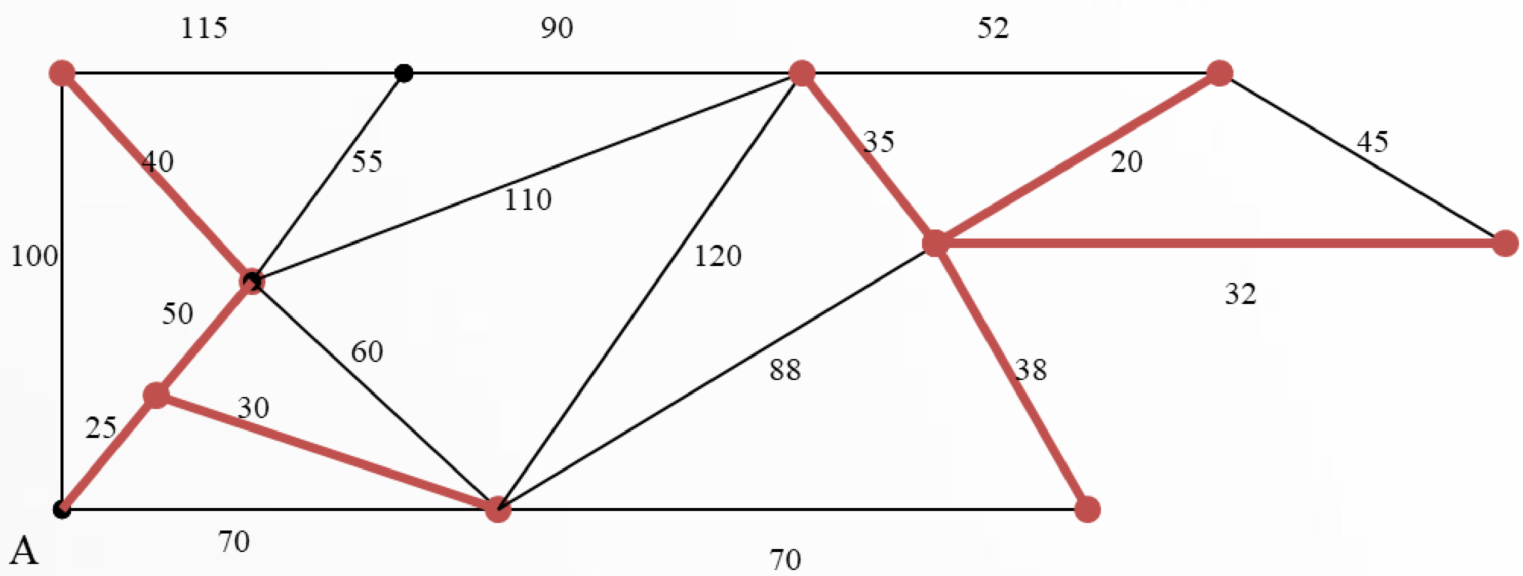
Kruskal's Algorithm



Add the next edge in the list to the tree if it does not close up a circuit with the edges chosen up to that point: 20, 25, 30, 32, 35, 38, 40, 45, 50, 52, 55, 60, 70, 70, 88, 90, 100, 110, 115, 120

Content Source: Discrete Mathematics and Its Applications. Seventh Edition. Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University. (and formerly AT&T Laboratories)

Kruskal's Algorithm

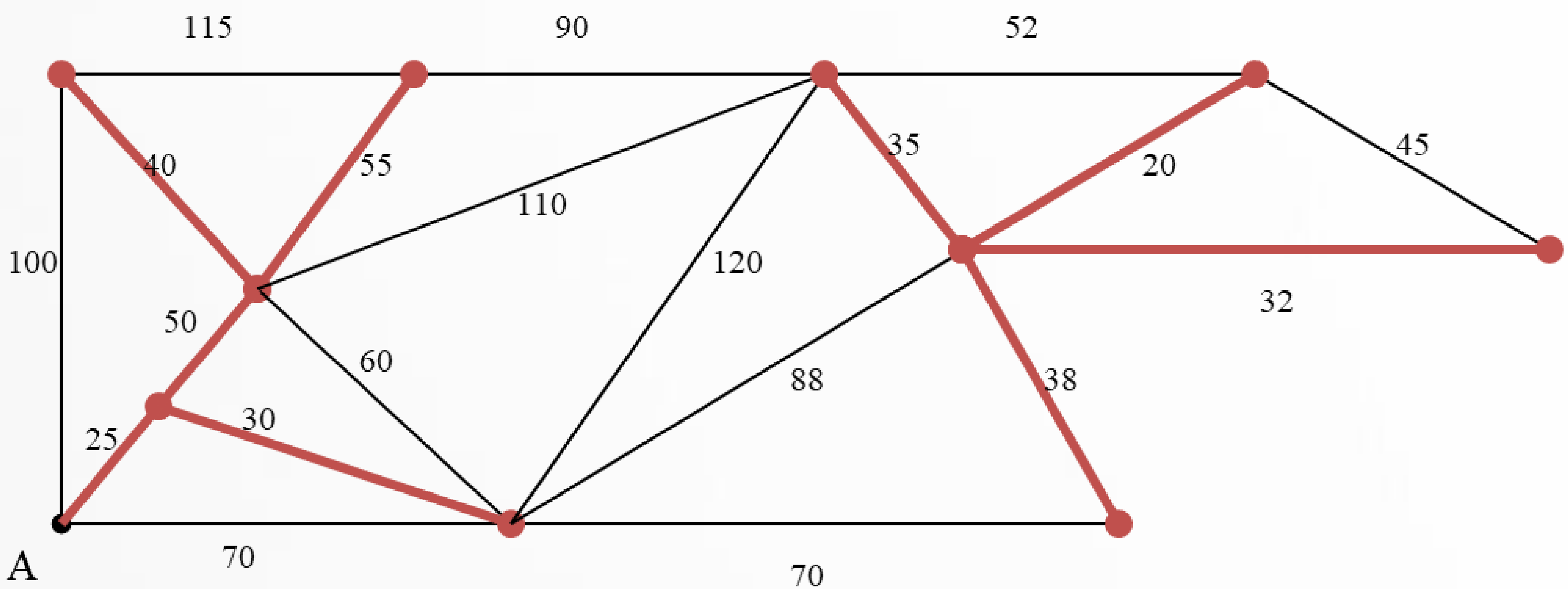


Add the next edge in the list to the tree if it does not close up a circuit with the edges chosen up to that point. Notice that the edge of weight 45 would close a circuit, so we skip it.

20, 25, 30, 32, 35, 38, 40, 45, 50, 52, 55, 60, 70, 70, 88, 90, 100, 110, 115, 120

Content Source: Discrete Mathematics and Its Applications. Seventh Edition. Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University. (and formerly AT&T Laboratories)

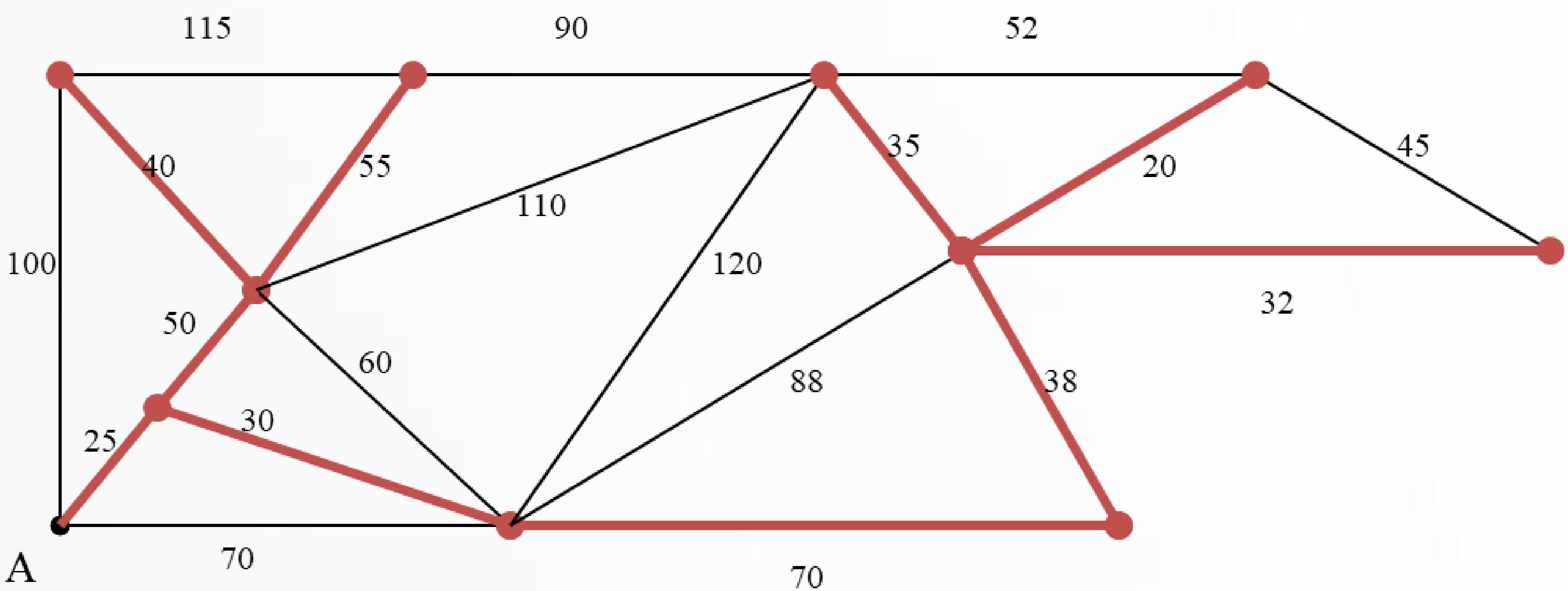
Kruskal's Algorithm



Add the next edge in the list to the tree if it does not close up a circuit with the edges chosen up to that point:

20, 25, 30, 32, 35, 38, 40, 45, 50, 52, 55, 60, 70, 70, 88, 90, 100, 110, 115, 120

Kruskal's Algorithm



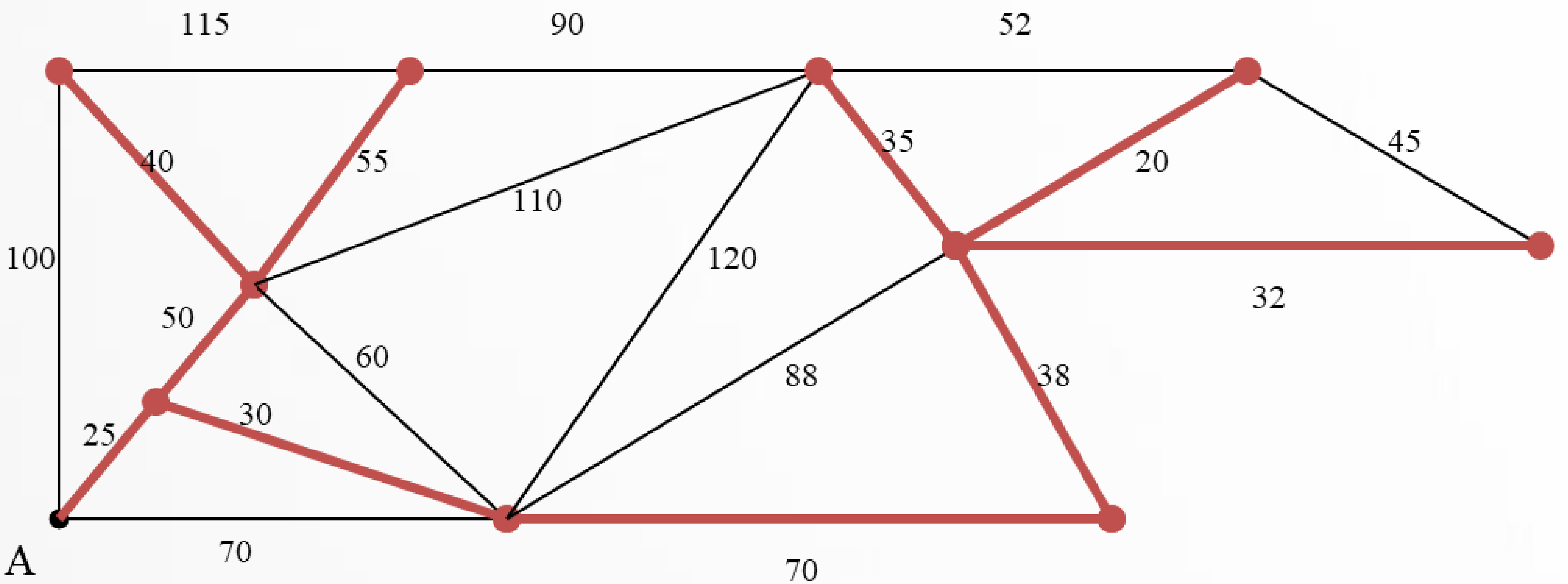
Add the next edge in the list to the tree if it does not close up a circuit with the edges chosen up to that point:

20, 25, 30, 32, 35, 38, 40, 45, 50, 52, 55, 60, 70, 70, 88, 90, 100, 110, 115, 120

Content Source: Discrete Mathematics and Its Applications. Seventh Edition. Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University. (and formerly AT&T Laboratories)

Kruskal's Algorithm

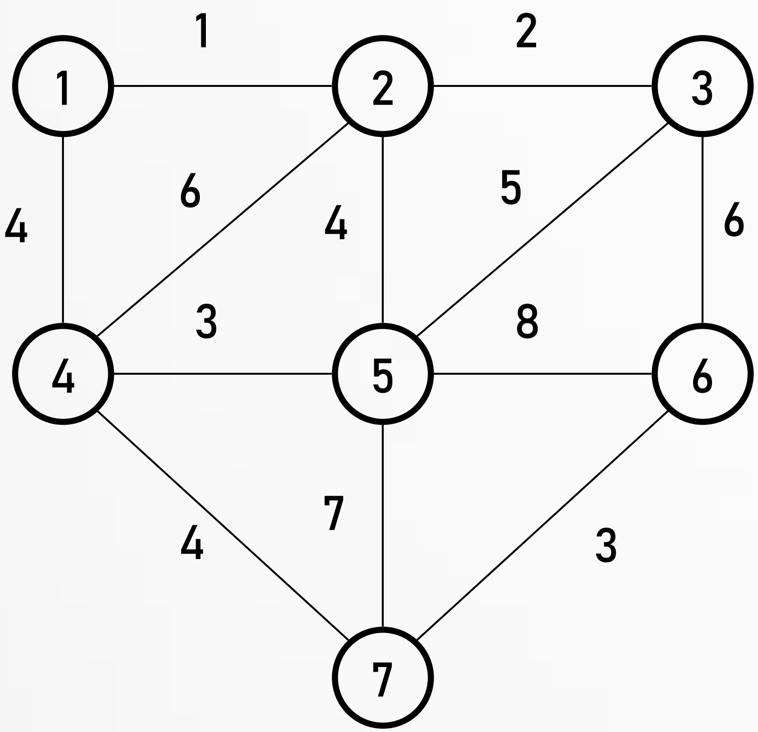
Done!



The tree contains every vertex, so it is a spanning tree. The total weight is 395

Content Source: **Discrete Mathematics** and **Its Applications**. Seventh Edition. **Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University.** (and formerly AT&T Laboratories)

Kruskal's Algorithm

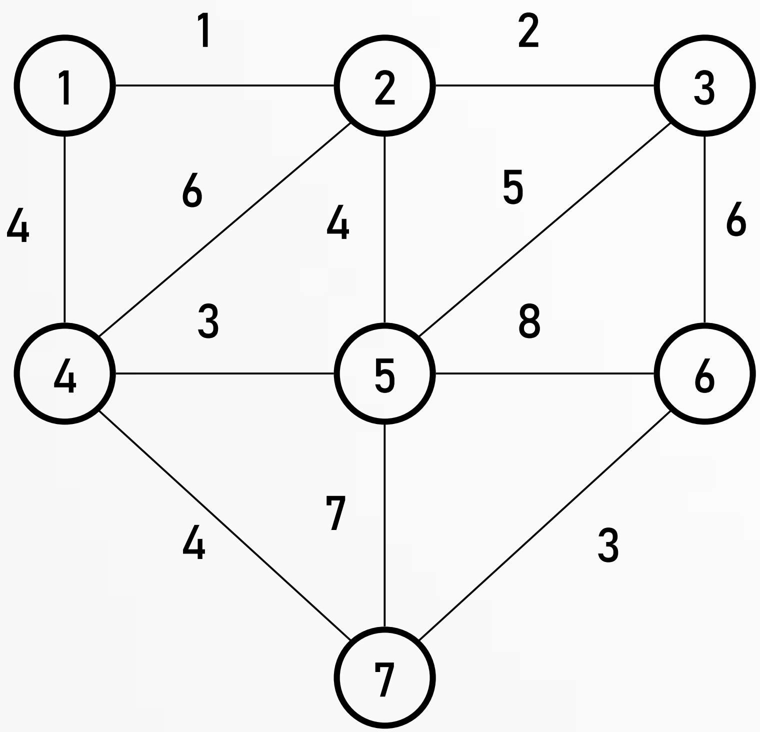


Make a disjoint set for each vertex

{1}{2}{3}{4}{5}{6}{7}

Content Source: **Discrete Mathematics** and **Its Applications**. Seventh Edition. **Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University.** (and formerly AT&T Laboratories)

Kruskal's Algorithm

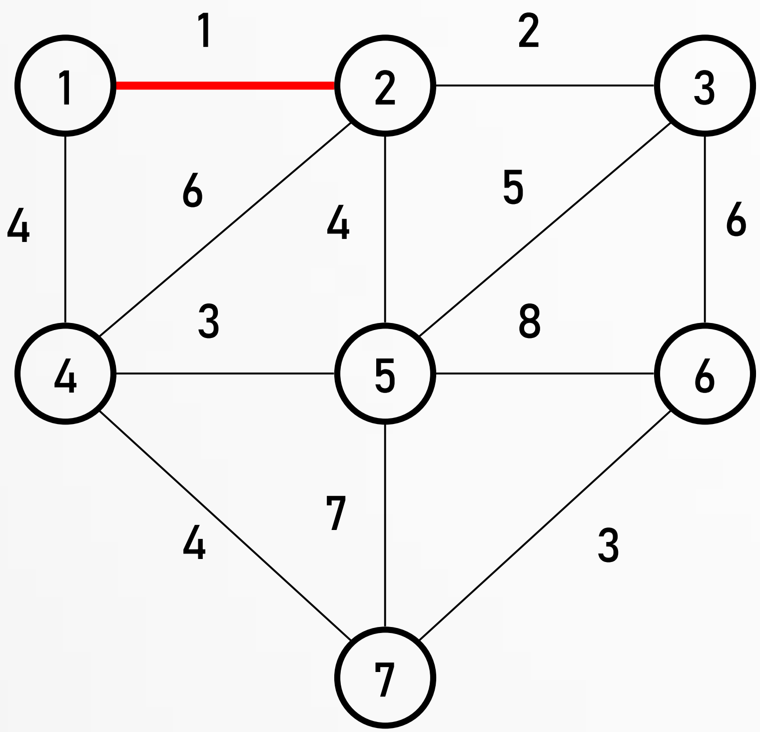


Sort edges by weight

1: {1,2}   {1}{2}{3}{4}{5}{6}{7}
2: {2,3}
3: {4,5}
3: {6,7}
4: {1,4}
4: {2,5}
4: {4,7}
5: {3,5}
⋮

Content Source: **Discrete Mathematics** and **Its Applications**. Seventh Edition. **Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University.** (and formerly AT&T Laboratories)

Kruskal's Algorithm

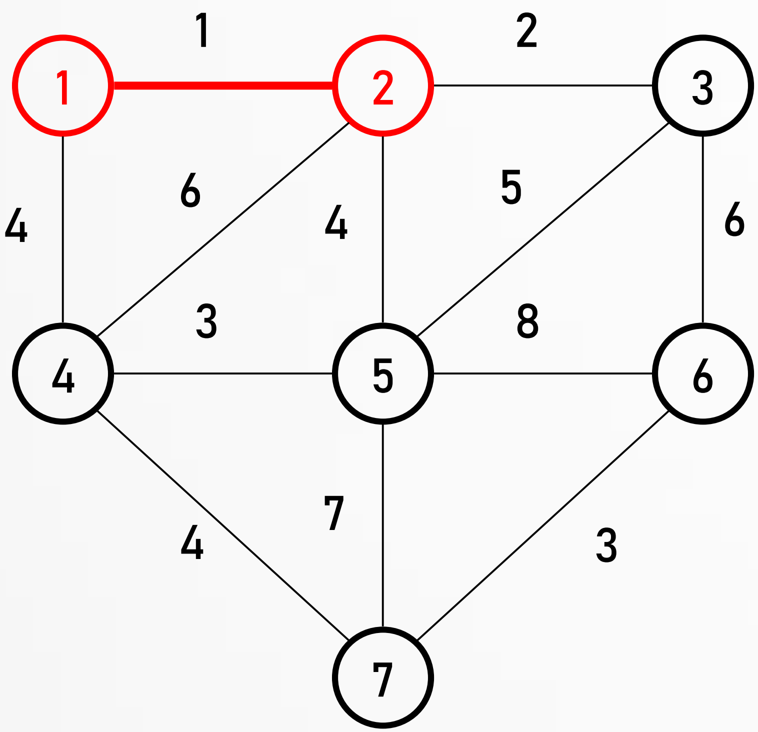


Add first edge to X if no cycle created

1: {1,2} {1}{2}{3}{4}{5}{6}{7}
2: {2,3}
3: {4,5}
3: {6,7}
4: {1,4}
4: {2,5}
4: {4,7}
5: {3,5}
⋮

Content Source: Discrete Mathematics and Its Applications. Seventh Edition. Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University. (and formerly AT&T Laboratories)

Kruskal's Algorithm

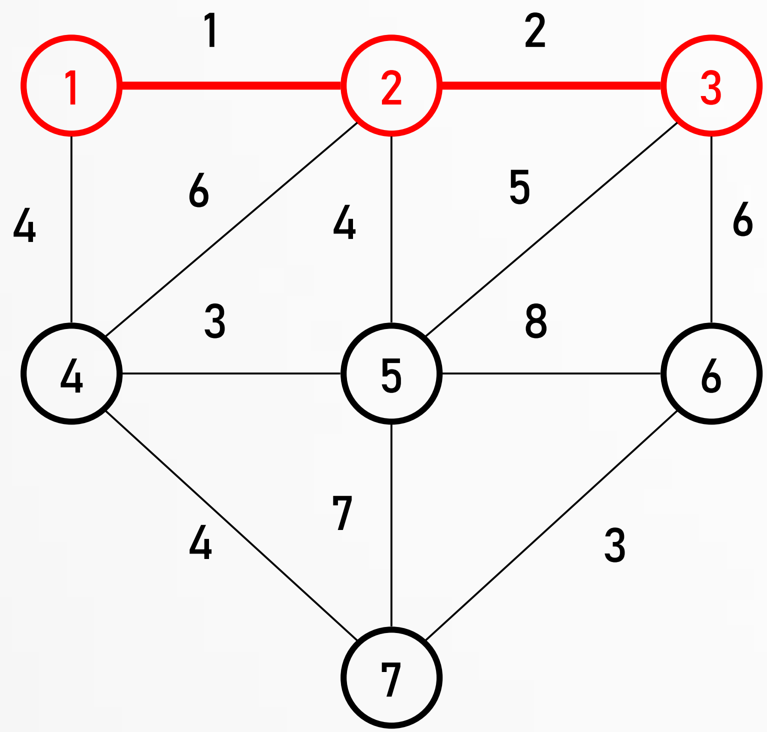


Merge vertices in added edges

1: {1,2}   {1,2}{3}{4}{5}{6}{7}
2: {2,3}
3: {4,5}
3: {6,7}
4: {1,4}
4: {2,5}
4: {4,7}
5: {3,5}
⋮

Content Source: Discrete Mathematics and Its Applications. Seventh Edition. Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University. (and formerly AT&T Laboratories)

Kruskal's Algorithm

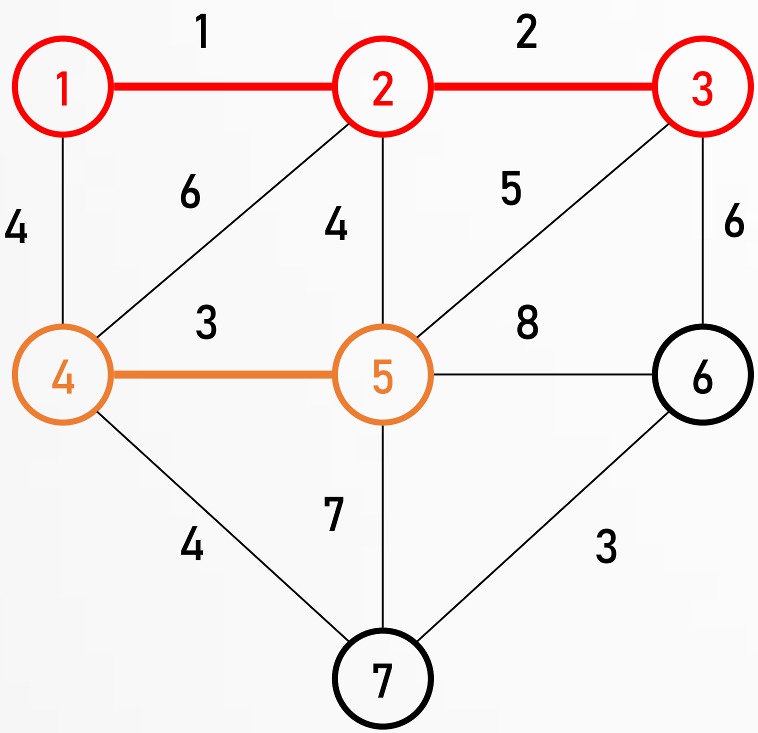


Process each edge in order

1: {1,2} {1,2}{3}{4}{5}{6}{7}
2: {2,3} {1,2,3}{4}{5}{6}{7}
3: {4,5}
3: {6,7}
4: {1,4}
4: {2,5}
4: {4,7}
5: {3,5}
⋮

Content Source: **Discrete Mathematics** and **Its Applications**. Seventh Edition. **Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University.** (and formerly AT&T Laboratories)

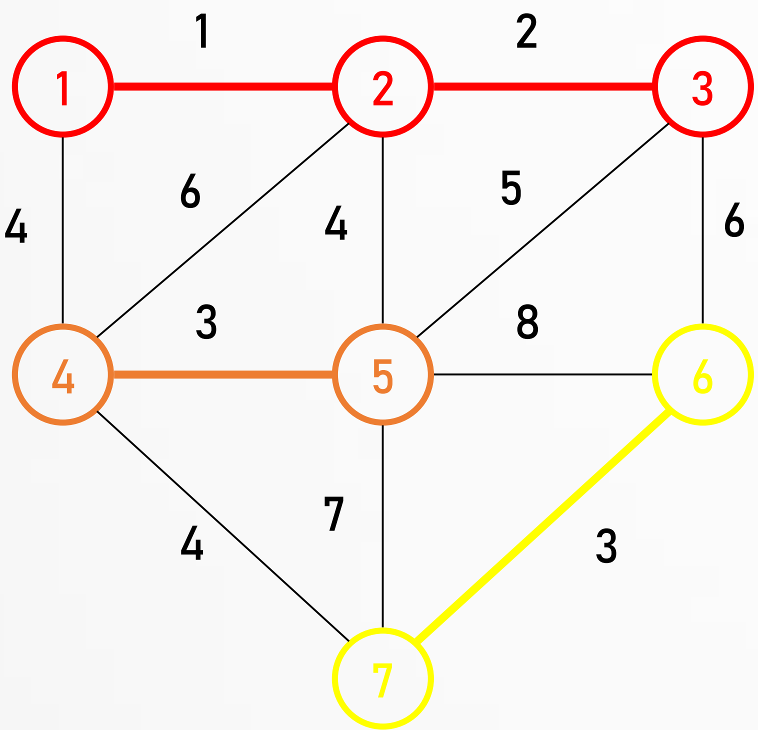
Kruskal's Algorithm



1: {1,2} {1,2}{3}{4}{5}{6}{7}
2: {2,3} {1,2,3}{4}{5}{6}{7}
3: {4,5} {1,2,3}{4,5}{6}{7}
3: {6,7}
4: {1,4}
4: {2,5}
4: {4,7}
5: {3,5}
⋮

Note that each set is a connected component of *G*

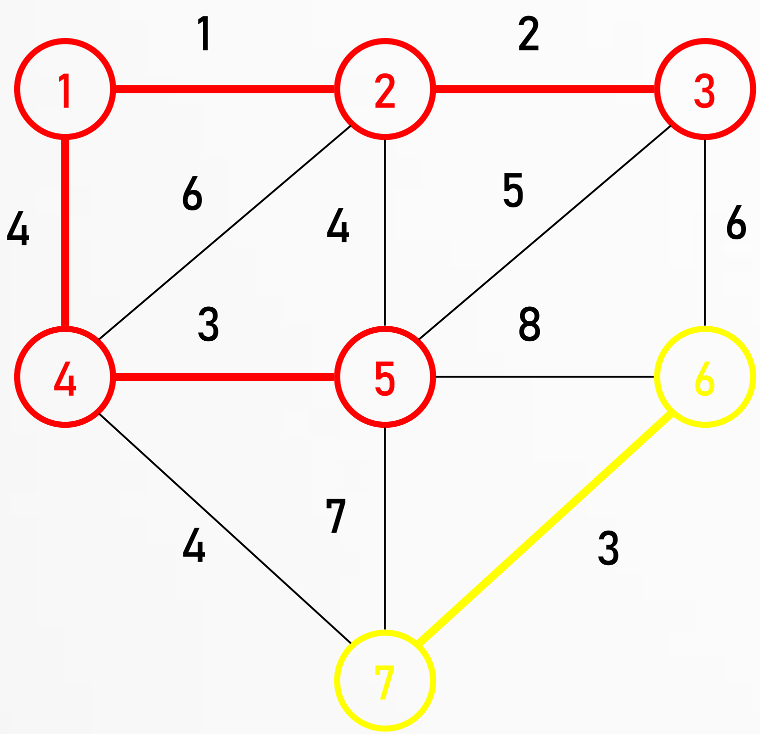
Kruskal's Algorithm



1: {1,2} {1,2}{3}{4}{5}{6}{7}
2: {2,3} {1,2,3}{4}{5}{6}{7}
3: {4,5} {1,2,3}{4,5}{6}{7}
3: {6,7} {1,2,3}{4,5}{6,7}
4: {1,4}
4: {2,5}
4: {4,7}
5: {3,5}
⋮

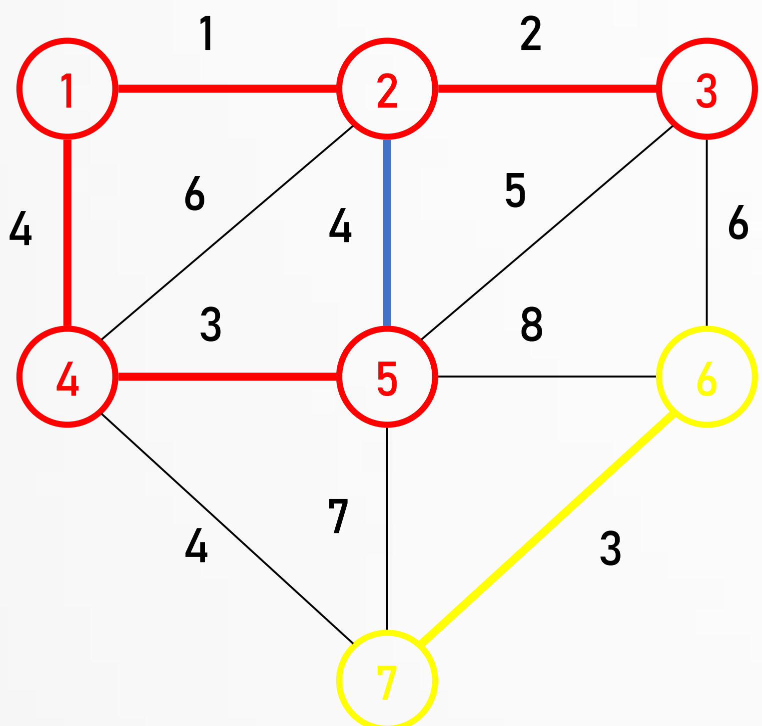
Content Source: **Discrete Mathematics** and **Its Applications**. Seventh Edition. **Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University.** (and formerly AT&T Laboratories)

Kruskal's Algorithm



1: {1,2} {1,2}{3}{4}{5}{6}{7}
2: {2,3} {1,2,3}{4}{5}{6}{7}
3: {4,5} {1,2,3}{4,5}{6}{7}
3: {6,7} {1,2,3}{4,5}{6,7}
4: {1,4} {1,2,3,4,5}{6,7}
4: {2,5}
4: {4,7}
5: {3,5}
⋮

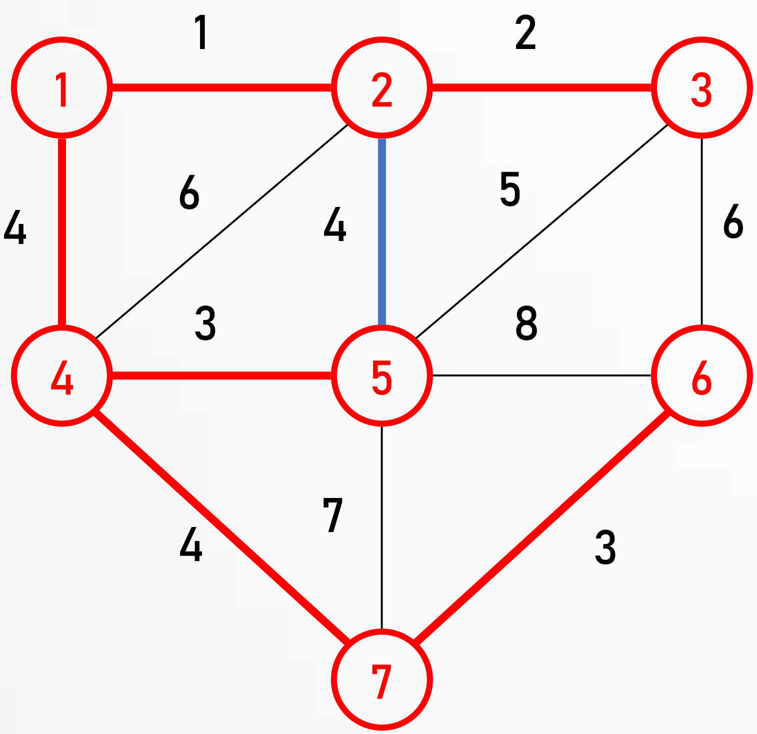
Kruskal's Algorithm



Must join separate components

1: {1,2} {1,2}{3}{4}{5}{6}{7}
2: {2,3} {1,2,3}{4}{5}{6}{7}
3: {4,5} {1,2,3}{4,5}{6}{7}
3: {6,7} {1,2,3}{4,5}{6,7}
4: {1,4} {1,2,3,4,5}{6,7}
4: {2,5} rejected
4: {4,7}
5: {3,5}
⋮

Kruskal's Algorithm



Done when all vertices in one set
Then they are all connected
Exactly |V| - 1 edges

1: {1,2} {1,2}{3}{4}{5}{6}{7}
2: {2,3} {1,2,3}{4}{5}{6}{7}
3: {4,5} {1,2,3}{4,5}{6}{7}
3: {6,7} {1,2,3}{4,5}{6,7}
4: {1,4} {1,2,3,4,5}{6,7}
4: {2,5} rejected
4: {4,7} {1,2,3,4,5,6,7} done
5: {3,5}
⋮

Content Source: **Discrete Mathematics** and **Its Applications**. Seventh Edition. **Kenneth H. Rosen. Monmouth University. Prepared by Jerrold W. Grossman, Oakland University.** (and formerly AT&T Laboratories)

That's all for now...