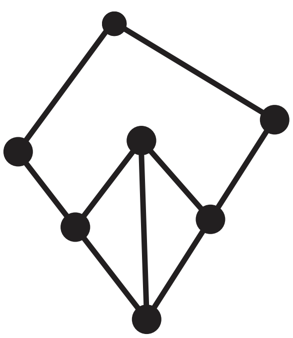
CHAPTER 2: BUSINESS EFFICIENCY

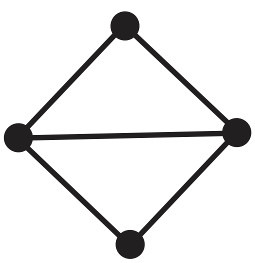
2.1 Hamiltonian Circuits

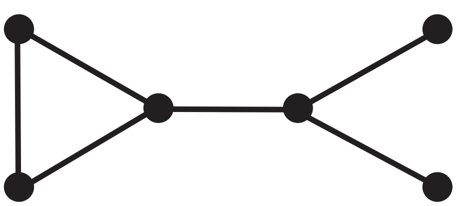
A path that visits each vertex exactly once is a ***Hamiltonian path***.

A circuit that visits each vertex exactly once is a ***Hamiltonian circuit***.

Example
Determine if the graphs below have a Hamiltonian path or circuit.



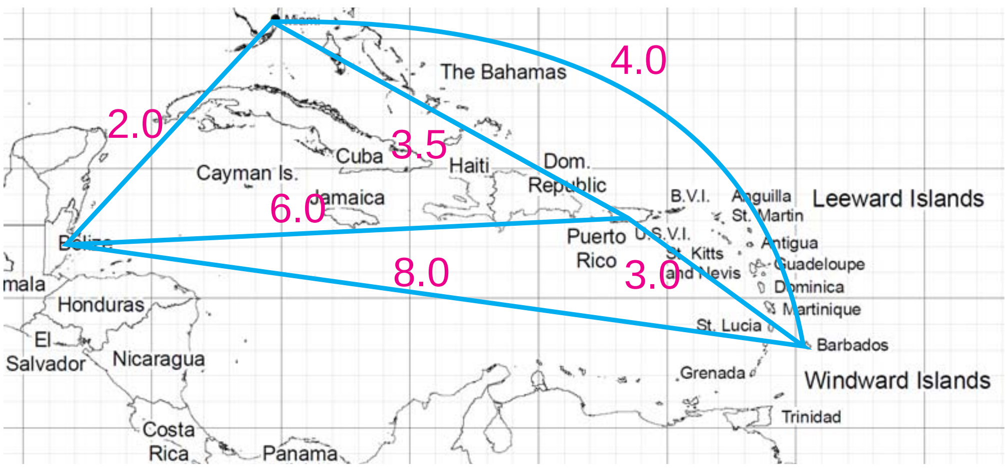




To find all possible Hamiltonian circuits, you can use the ***method of trees*** to list all possible paths from a particular starting point.

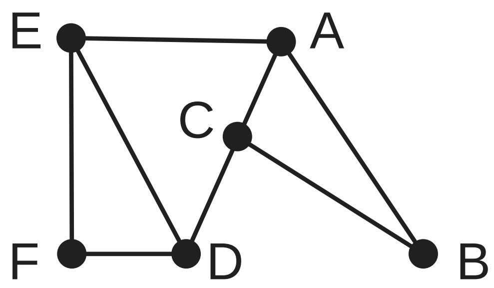
An ***optimal*** Hamiltonian circuit is the circuit that takes the least weight.

<u>*Example*</u>
Represent the data in the map below in a weighted graph. Then use the method of trees to determine the minimum cost (travel time, in hours) to visit Miami, San Juan, Barbados and Belize City if you must start and end in Miami. In what order do you visit the cities?



Example

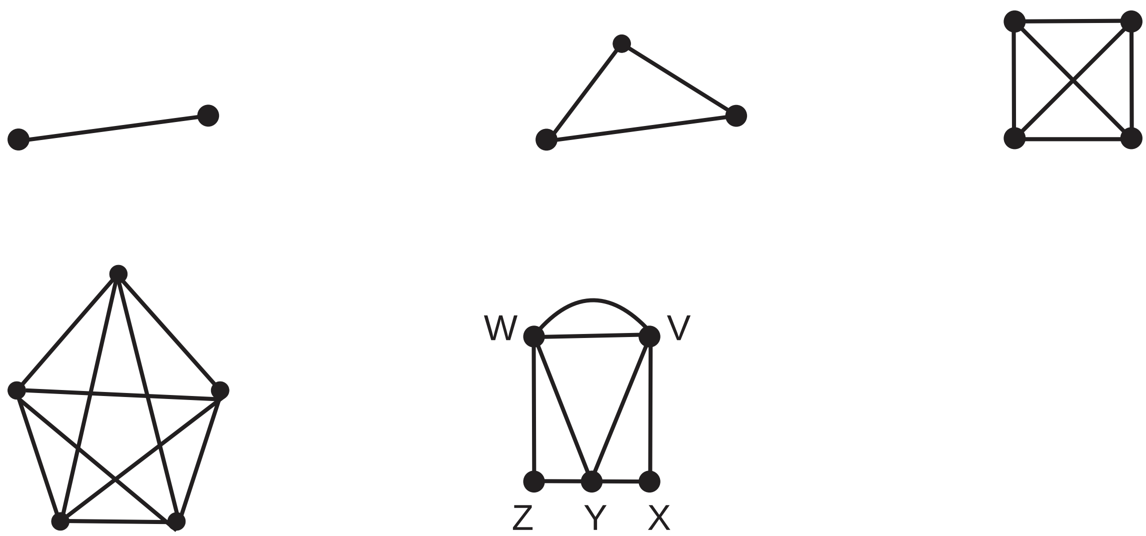
Use the method of trees to find all Hamiltonian circuits starting at A.



A ***complete*** graph is a graph in which every pair of vertices is connected by exactly one edge.

Example

Determine which of the graphs below are complete.



How many edges does a complete graph with *v* vertices have?

Fundamental Theorem of Counting:

Suppose you have *k* tasks to be performed. The first task can be completed n_1 ways, the second task n_2 ways, etc. The total number of ways that these *k* tasks can be performed is the product

$$n_1 \times n_2 \times \ldots n_k$$

Brute Force Method:

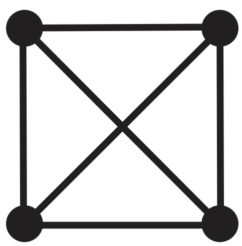
- There are $n!$ routes that visit every vertex once in a complete graph.
- There are $n!/2$ unique Hamiltonian circuits in a complete graph.
- If a starting point is specified, there are $(n-1)!/2$ unique Hamiltonian circuits in a complete graph.

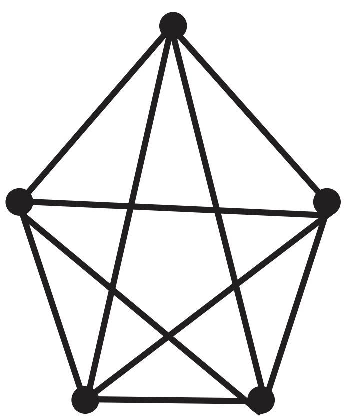
2.2 Traveling Salesman Problem

The traveling salesman problem is a least cost Hamiltonian circuit problem.

Example

Use the brute-force method to find all unique Hamiltonian circuits for the complete graphs below starting at A.





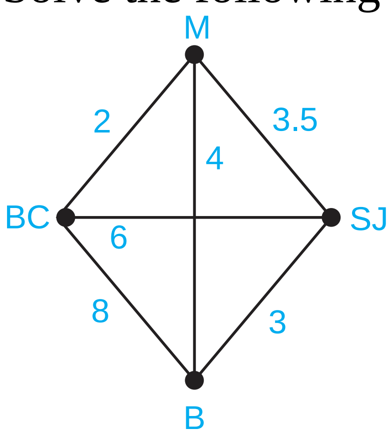
The TSP is an important and common problem to solve, so we need ***heuristic algorithms***. These are algorithms that are fast but may not be optimal.

2.3 Helping Traveling Salesmen

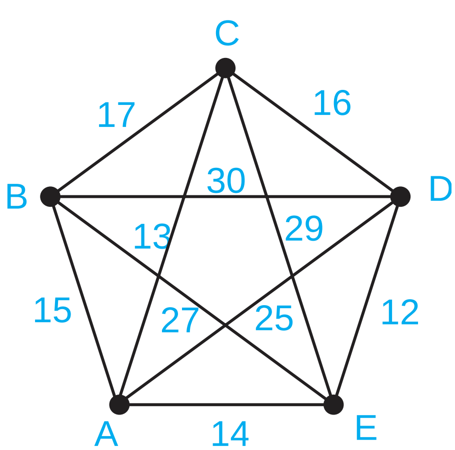
Nearest Neighbor Algorithm:
Starting from the home city, visit the nearest city first. Then visit the nearest city that has not already been visited. Return to the home city when no other choices remain.

Note that a ***greedy algorithm*** is one in which the choices are made by what is best at the next step.

Example
Solve the following TSP using the NN algorithm starting at M.



Example
Solve the following TSP using the NN algorithm starting at D and also starting at A. Values are distances between points in miles.



Example

Solve the following TSP using the NN algorithm starting at Afton. If the brute force method was used, how many unique circuits would need to be checked? The distance is in miles.

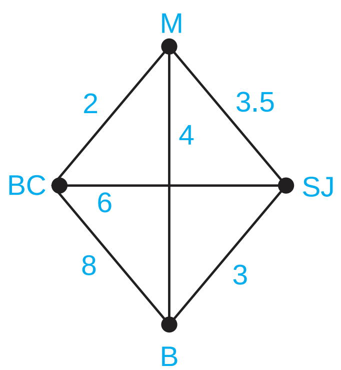
	Afton	**Bar Nunn**	**Casper**	**Laramie**	**Newcastle**	**Pine Bluffs**	**Rock Springs**	**Sleepy Hollow**
Afton	0	355	356	387	537	477	180	477
Bar Nunn		0	7	146	189	227	228	122
Casper			0	142	185	223	228	128
Laramie				0	246	89	207	249
Newcastle					0	217	409	78
Pine Bluffs						0	296	285
Rock Springs							0	351
Sleepy Hollow								0

Sorted Edges Algorithm:

1. Arrange edges of the complete graph in order of increasing cost
2. Select the lowest cost edge that has not already been selected that
   a. Does not cause a vertex to have 3 edges
   b. Does not close the circuit unless all vertices have been included.

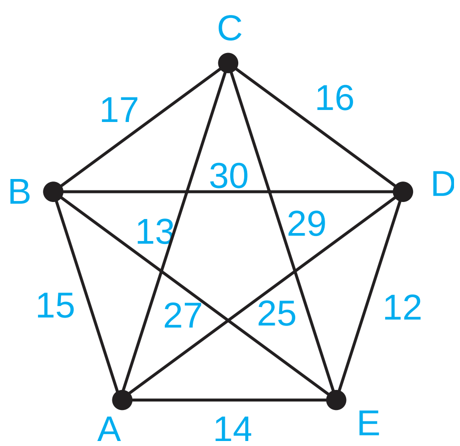
<u>*Example*</u>

Solve the following TSP using the SE algorithm.



<u>*Example*</u>

Solve the following TSP using the SE algorithm.



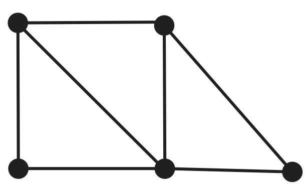
2.4 Minimum Cost Spanning Trees

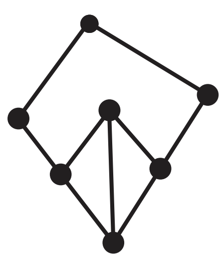
A connected graph that has no circuits is a ***tree***. A ***spanning tree*** is a tree that has all the vertices of the original graph.

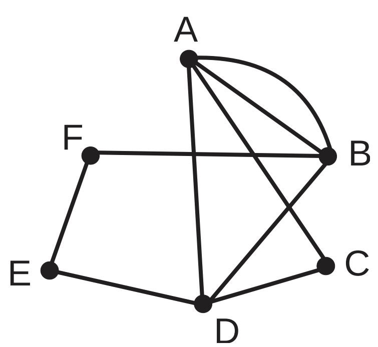
To create a spanning tree from a graph,

1. Find a circuit and remove one edge
2. Continue until there are no circuits

Example
Remove the edges from the following graphs to form a subgraph that is a tree.



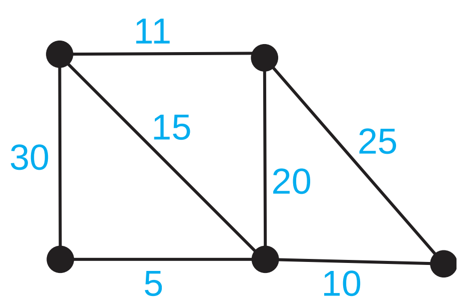




A ***minimal spanning tree*** is a spanning tree with the smallest possible weight.

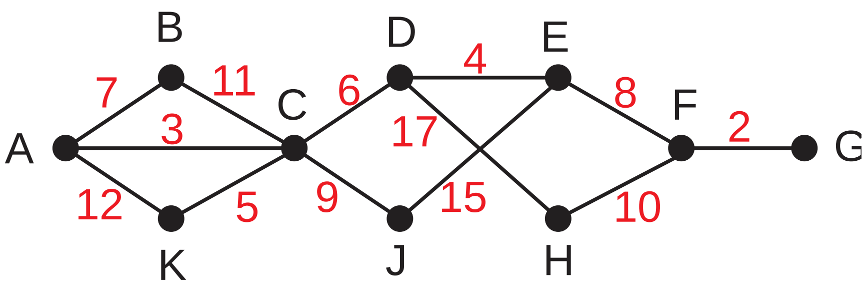
Kruskal's Algorithm:
Add edges in order of increasing cost so that no circuit is formed.

<u>*Example*</u>
Use Kruskal's Algorithm to find the minimal spanning tree from the graph below:

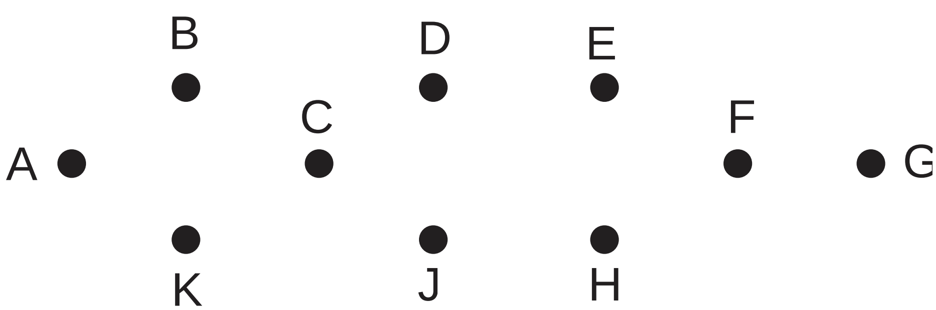


<u>*Example*</u>

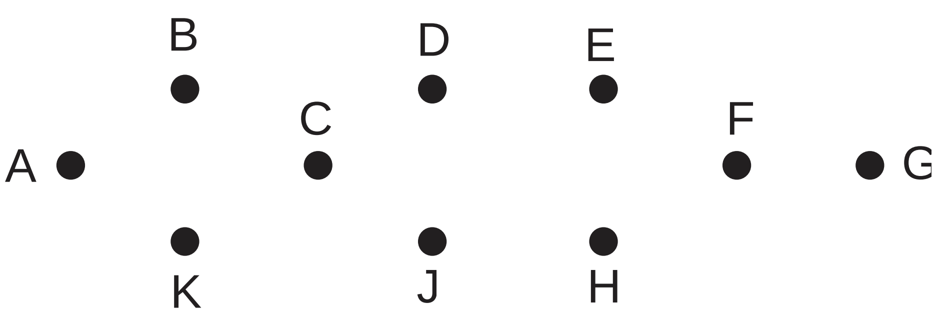
Use the weighted graph below to find a minimal and a maximal spanning tree. What are some possible applications of a maximal spanning tree?



Minimal spanning tree has a total cost of



Maximal spanning tree has a total cost of



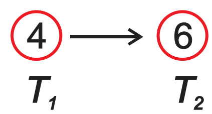
2.5 Critical Path Analysis

A list of vertices connected by arrows is a ***directed graph*** or ***digraph***.

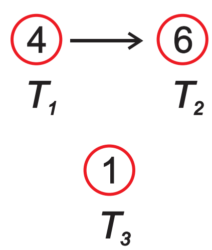
If the tasks cannot be completed in a random order, then the order can be specified in an ***order-requirement digraph***.

If the time to complete a task is shown on the digraph, it is a ***weighted*** digraph.

Suppose the first task T_1 takes 4 minutes and a second task T_2 takes 6 minutes and the second task can't be started until the first task is done. This would be represented in a weighted digraph as



An ***independent task*** is one that can be done independently of any of the other tasks. So if task T_3 takes 1 minute and an independent task, the weighted order-requirement digraph would look like



Example

To make a tea tray requires you to complete the following tasks

T_1: get the tea leaves (3 minutes)
T_2: boil water (7 minutes)
T_3: brew tea (5 minutes)
T_4: pour milk (2 minutes)
T_5: fill sugar bowl (1 minute)
T_6: place items on tray (4 minutes)

Show this in a weighted order-requirement digraph.

Are any of the tasks independent?

How long would it take to prepare a tea tray if only one person was available? How might the person complete the tasks?

How long if two people were available? How might the people complete the tasks?

Example

Break down the recipe below into a series of tasks. Show these tasks in a weighted order-requirement digraph

Tortillas Chicken 1 onion Seasoning Tomatoes Cilantro	Slice the onion and chicken into strips. Cook chicken 10 minutes then add onions and cook for 5 more minutes. Add seasoning. Chop tomatoes and mix with cilantro. Warm tortillas. Serve

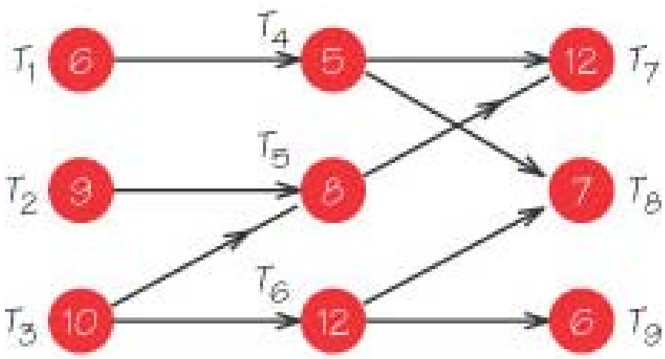
A ***critical path*** on the digraph is the path that determines the earliest completion time.

<u>*Example*</u>

What is the critical path for preparing tea?

What is the critical path for making fajitas?

Determine the critical path in the digraph below:



(b)