CHAPTER 2: BUSINESS EFFICIENCY

2.1 Hamiltonian Circuits

A path that visits each vertex exactly once is a ______________ path.

A circuit that visits each vertex exactly once (except the beginning point will be visited again) is a ________________ circuit.

Example
Determine if the graphs below have a Hamiltonian path or circuit.
To find all possible Hamiltonian circuits, you can use the ____________________ to list all possible paths from a particular starting point.

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

![Diagram of Hamiltonian circuit](image-url)
An ______________ Hamiltonian circuit is the circuit that has the least weight.

**Example**

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 (Puerto Rico), Barbados and Belize City (Belize) if you must start and end in Miami. In what order do you visit the cities?
A ___________ 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 $n$ 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

$$n_1 \times n_2 \times ... n_k$$

**Examples**
How many outfits can be made from 2 shirts and 5 pairs of pants, if an outfit consists of a shirt and a pair of pants?

How many different ways can 5 people stand in a line?
**Brute Force Method for a Complete Graph with n Vertices:**

- There are \( n! \) Hamiltonian circuits in a complete graph.
- There are \( \frac{n!}{2} \) Hamiltonian circuits in a complete graph, if a circuit and its mirror image are not counted as separate circuits.
- If a starting point is specified, there are \( (n - 1)! \) Hamiltonian circuits in a complete graph.
- If a starting point is specified, there are \( \frac{(n-1)!}{2} \) Hamiltonian circuits in a complete graph, if a circuit and its mirror image are not counted as separate circuits.

**Example**

Use the brute force method to find all different (non-mirror image) Hamiltonian circuits for the complete graphs below *starting at A*.
Example
Use the brute force method to find all different (non-mirror image) Hamiltonian circuits for the complete graphs below starting at A.
2.2 Traveling Salesman Problem (TSP)

The traveling salesman problem is the problem of finding a least cost Hamiltonian circuit in a complete graph where each edge has been assigned a cost (or weight).

*Example*
Use the brute force method to find the least cost Hamiltonian circuit, starting at A, in the graph below. The values are distances between points, in miles.

![Graph of the Traveling Salesman Problem](image)

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

**Nearest Neighbor (NN) Algorithm** (for finding low-cost Hamiltonian circuits):
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 ____________ algorithm is one in which the choices are made by what is best at the next step.

**Example**
Use the NN algorithm to solve the TSP of finding a low-cost trip starting at Miami (M) and traveling to the other three places. The weights given are the travel times between places, in hours.)

Is the solution you found the same as the optimal solution found when we used brute force?
Example
Use the NN algorithm to find a low-cost Hamiltonian circuit (solve the TSP) given the graph below, starting at D and also starting at A. The values are distances between points, in miles.

Starting at D:

Starting at A:

Is the solution you found starting at A the same as the optimal solution found when we used brute force?
Example

The following table gives the distances between the different cities, in miles.

<table>
<thead>
<tr>
<th></th>
<th>Afton</th>
<th>Bar Nunn</th>
<th>Casper</th>
<th>Laramie</th>
<th>Newcastle</th>
<th>Pine Bluffs</th>
<th>Rock Springs</th>
<th>Sleepy Hollow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afton</td>
<td>0</td>
<td>355</td>
<td>356</td>
<td>387</td>
<td>537</td>
<td>477</td>
<td>180</td>
<td>477</td>
</tr>
<tr>
<td>Bar Nunn</td>
<td>355</td>
<td>0</td>
<td>7</td>
<td>146</td>
<td>189</td>
<td>227</td>
<td>228</td>
<td>122</td>
</tr>
<tr>
<td>Casper</td>
<td>356</td>
<td>7</td>
<td>0</td>
<td>142</td>
<td>185</td>
<td>223</td>
<td>228</td>
<td>128</td>
</tr>
<tr>
<td>Laramie</td>
<td>387</td>
<td>146</td>
<td>142</td>
<td>0</td>
<td>246</td>
<td>89</td>
<td>207</td>
<td>249</td>
</tr>
<tr>
<td>Newcastle</td>
<td>537</td>
<td>189</td>
<td>185</td>
<td>246</td>
<td>0</td>
<td>217</td>
<td>409</td>
<td>78</td>
</tr>
<tr>
<td>Pine Bluffs</td>
<td>477</td>
<td>227</td>
<td>223</td>
<td>89</td>
<td>217</td>
<td>0</td>
<td>296</td>
<td>285</td>
</tr>
<tr>
<td>Rock Springs</td>
<td>180</td>
<td>228</td>
<td>228</td>
<td>207</td>
<td>409</td>
<td>296</td>
<td>0</td>
<td>351</td>
</tr>
<tr>
<td>Sleepy Hollow</td>
<td>477</td>
<td>122</td>
<td>128</td>
<td>249</td>
<td>78</td>
<td>285</td>
<td>351</td>
<td>0</td>
</tr>
</tbody>
</table>

If the brute force method was used to find the least cost Hamiltonian circuit (solve the TSP) starting at Afton, how many different (non-mirror image) circuits would need to be checked?

Use the NN algorithm to solve the TSP, starting at Afton.

Use the NN algorithm to solve the TSP, starting at Bar Nunn.
Sorted Edges (SE) Algorithm (for finding low-cost Hamiltonian circuits):

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.

Example
Solve the following TSP using the SE algorithm. (Remember this graph represents places and the travel times between them, in hours.)

Is the solution you found optimal?

Example
Solve the following TSP using the SE algorithm. (Remember the weights given are distances, in miles.)

Is the solution you found optimal?
2.4 Minimum Cost Spanning Trees

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

To create a spanning tree from a graph,
1. Copy the vertices with no edges
2. Add edges back one by one until you have a connected graph that uses all vertices and contains no circuits

Example
For the following graphs, form a subgraph that is a spanning tree.
A ______________spanning tree__ is a spanning tree with the smallest possible weight.

**Kruskal’s Algorithm for finding minimum-cost spanning trees:**
Finding a minimum-cost spanning tree by adding edges in order of increasing cost so that no circuit is formed.

**Example**
Use Kruskal’s Algorithm to find a minimum-cost spanning tree from the graphs below (in each, the weights given are times, in minutes):

Minimum-cost spanning tree has a total cost of _________________.

Minimum-cost spanning tree has a total cost of _________________.

2.5 Critical Path Analysis

A list of vertices connected by arrows is a __________ graph or __________.

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

If the time to complete a task is shown on the digraph, it is a ______________ 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

![Weighted digraph](image)

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

![Weighted digraph](image)
Example
Break down the fajita recipe below into a series of tasks. Show these tasks in a weighted order-requirement digraph.

Tortillas  Slice the onion and chicken into strips.
Chicken    Cook chicken 10 minutes then add
1 onion    onions and cook for 5 more minutes.
Seasoning  Add seasoning. Chop tomatoes and mix
Tomatoes   with cilantro. Warm tortillas. Serve.
Cilantro
A ____________ path on the digraph is the longest path and it determines the earliest completion time (the earliest possible time for the completion of all of the tasks making up the job in the digraph).

*Example*
What is the critical path for making fajitas?

*Example*
Determine the critical path in the digraph below.

What is the earliest possible time for completion of all of the tasks in the digraph?