12.3 Notes: Hamilton Paths and Hamilton Circuits

Objectives

1. **Can you define Hamilton paths & Hamilton circuits?**
2. **Can you find the number of Hamilton circuits in a complete graph?**
3. **Can you use weighted graphs to solve problems?**

Vocabulary:

- A path that passes through each vertex of a graph exactly once is called a Hamilton Path.
- If a Hamilton path begins and ends at the same vertex and passes through all other vertices exactly once, it is called a Hamilton Circuit.
- A complete graph is a graph that has an edge between each pair of its vertices.

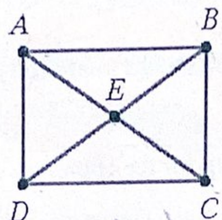
complete (3 or more) ⇒ Hamilton Circuit

Examples:

1) Find a Hamilton Path.

A, B, C, D, E

(No need to use every edge – just every vertex)

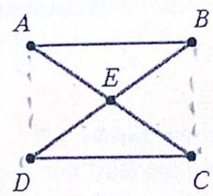


2) Find a Hamilton Circuit.

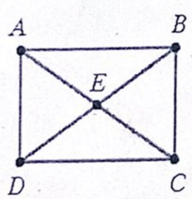
A, B, C, D, E, A

3) Is the graph shown connected? Complete?

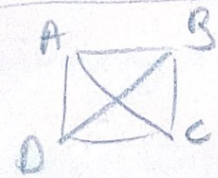
Yes No



4) Is the graph below connected? Complete?



Yes No – No AC or BD



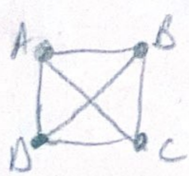
is complete

Note: The number of Hamilton circuits in a complete graph with n vertices is $(n-1)!$.

$(5-1)! = 4! = 4\cdot3\cdot2\cdot1 = 24$

5) How many Hamilton circuits are there in the graph from #4? List them!

Let's do this one instead:



✓ Complete graph

4 vertices:

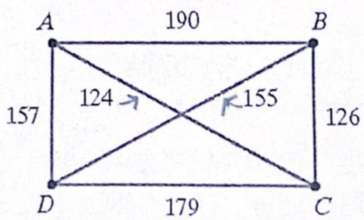
$(4-1)!$

$= 3! = 3\cdot2\cdot1 = \boxed{6}$

A, B, C, D, A
A, B, D, C, A
A, C, B, D, A
A, C, D, B, A
A, D, C, B, A
A, D, B, C, A

Weighted Graphs

- A complete graph whose edges have numbers attached to them is called a weighted graph.
- The numbers shown along the edges of a weighted graph are called the *weights* of the edges.



Examples: The graph shown is a model graph for one-way airfares for cities *A*, *B*, *C*, *D*.

1) Find the weight of edge AC. 124

2) Use the weighted graph to find the cost of the trip for the Hamilton circuit *A*, *B*, *D*, *C*, *A*. 190 + 155 + 179 + 124 = #648

3) Use the weighted graph to find the cost of the trip for the Hamilton circuit *A, D, C, B, A*.

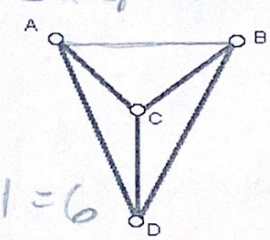
157 + 179 + 126 + 190 = #652

4) Use the weighted graph to find the cost of the trip for the Hamilton circuit *A, C, B, D, A*.

124 + 126 + 155 + 157 = #562

5) Which Hamilton circuit would be the least expensive?

#4) A, C, B, D, A → Traveling Salesperson Prob



Example: Use the graph shown for # 6 – 8.

6) Modify the graph by adding the least number of edges so that the resulting graph is complete. Determine the number of Hamilton Circuits for the graph.

add AB

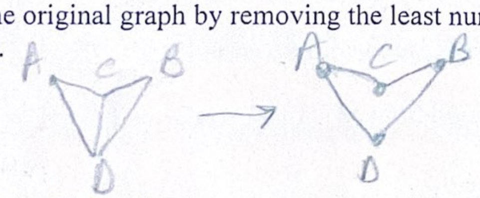
$(n-1)! = (4-1)! = 3! = 3 \cdot 2 \cdot 1 = 6$

7) Give two Hamilton Circuits for your new graph. A, B, C, D, A

A, C, D, B, A

8) Modify the original graph by removing the least number of edges so that the resulting graph has an Euler Circuit.

all even vertices



9) Find an Euler Circuit for the graph drawn in #8. travel every edge

A, C, B, D

12.3 Continued: Methods of Solving Traveling Salesperson Problems

Objectives

1. Can you use the Brute Force Method to solve traveling salesperson problems?
2. Can you use the Nearest Neighbor Method to approximate solutions to traveling salesperson problems?
3. Can you describe the benefits and weaknesses of the Nearest Neighbor Method?

The Traveling Salesperson Problem (TSP) → every vertex

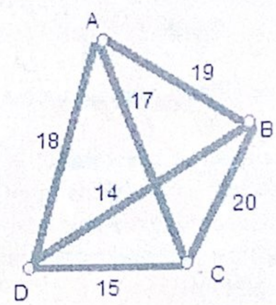
The *traveling salesperson problem* is the problem of finding a __Hamilton__ circuit in a complete weighted graph for which the sum of the weights of the edges is a __minimum__.

Note: Such a Hamilton circuit is called the *optimal Hamilton circuit* or the *optimal solution.*

The Brute Force Method: One way to find the optimal solution for a TSP.

- Model the problem with a __complete__, weighted graph.
- Make a __list__ of all possible Hamilton circuits.
- Determine the __sum__ of the weights of the edges for each of these Hamilton circuits.
- The Hamilton circuit with the __min.__ sum of weights is the optimal solution.

Example: A businessman lives in City D and needs to travel to cities A, B, and C before returning home. The weighted graph shows the cost (in dollars) of gasoline used to drive to each city. Use the Brute Force Method to find his optimal circuit.



Hamilton Circuit	Weights	Sum of Weights
A, B, C, D, A	19 + 20 + 15 + 18	72
A, B, D, C, A	19 + 14 + 15 + 17	65
A, C, B, D, A	17 + 20 + 14 + 18	69
A, C, D, B, A	17 + 15 + 14 + 19	65
A, D, B, C, A	18 + 14 + 20 + 17	69
A, D, C, B, A	18 + 15 + 20 + 19	72

The optimal circuit is __A B D C A__, with a weight of __65__.

OR A C D B A

Although the Bruce Force Method always provides the optimal solution, it is time-consuming and can be difficult for more difficult problems.
Reminder: for a complete graph, the number of possibilities can be found by using (n – 1)!
How many circuits would need to be considered in order to find the optimal solution to a TSP if there are 5 vertices? 6 vertices?

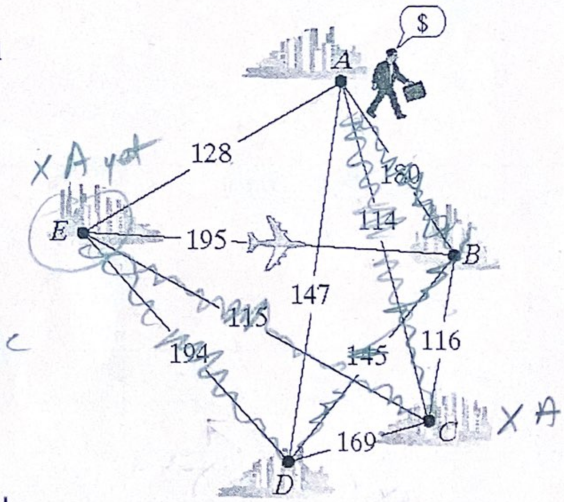
$(5-1)! = 4!$
$= 4 \cdot 3 \cdot 2 \cdot 1$
$= 24$

$(6-1)! = 5!$
$= 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$
$= 120$

The Nearest Neighbor Method: Used to approximate solutions to a TSP.

1. Model the problem with a complete, weighted graph.
2. Identify the vertex that serves as the starting point.
3. From the starting point, choose an edge with the smallest weight. **Darken** this edge to the second vertex.
4. From the 2nd vertex, choose the edge with the smallest weight that does not lead to a vertex already visited. **Darken** this edge to the third vertex.
5. Continue building the circuit, one vertex at a time, by moving along the edge with the smallest weight until all vertices are visited.
6. From the last vertex, return to the starting point.

Example: A sales director who lives in city *A* is required to fly to regional offices in cities *B*, *C*, *D*, and *E*. The weighted graph shown gives the one-way airfares. Use the Nearest Neighbor Method to find an approximate solution. What is the total cost?



A, C, E, D, B, A
114 + 115 + 194 + 145, 180
= $\boxed{\$748}$
forced into expensive one

Is this always going to be the optimal solution? No

How could the optimal solution be found by using the Nearest Neighbor Algorithm?
Repeat process with starting points

Why would the Nearest Neighbor Algorithm be used instead of the Brute Force method?
Many vertices – much quicker to compute