
MATH 11008: Brute-Force and Nearest Neighbor Algorithms

Section 6.5

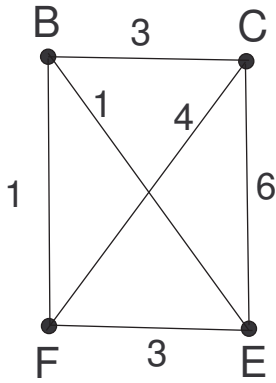
- **Traveling Salesman Problem:** Any problem that has a traveler, a set of sites, a cost function for travel between sites (weights on the edges), and need to tour all the sites (Hamilton circuit), and a desire to minimize the total cost of the tour (Hamilton circuit of least total weight) is known as a **traveling salesman problem** (TSP).
- **total weight of a circuit:** The **total weight** of a circuit is the sum of the weights on the edges of the circuit.
- **Minimum Hamilton circuit:** In a weighted graph, a **minimum Hamilton circuit** is a Hamilton circuit with smallest possible total weight.

Brute Force Algorithm:

1. Make a list of all the possible Hamilton circuits of the graph. Each of these circuits represents a tour of the vertices of the graph.
2. For each tour calculate its weight.
3. Choose a Hamilton circuit with smallest total weight. (There is always more than one optimal tour to choose from.)

- The Brute Force Algorithm is an example of an **optimal algorithm** because when implemented correctly it is guaranteed to produce an optimal solution.
- The Brute Force Algorithm is also an example of an **inefficient algorithm** because the number of steps needed to carry it out grows disproportionately with the number of vertices in the graph.

Example 1: Find a minimum Hamilton circuit for the complete, weighted graph shown below:



Nearest Neighbor Algorithm:

1. Start at the designated starting vertex. If there is no designated starting vertex pick any vertex.
2. From the starting vertex go to its nearest neighbor (the vertex for which the corresponding edge has the smallest weight).
3. From each vertex go to its nearest neighbor, choosing only among the vertices that haven't been yet visited. (If there is more than one nearest neighbor choose among them at random.) Keep doing this until all the vertices have been visited.
4. From the last vertex return to the starting vertex.

- The Nearest Neighbor Algorithm is also an example of an **efficient algorithm** because the amount of computational effort required to implement the algorithm grows in some reasonably proportion with the size of the input to the problem.
- Unfortunately, the Nearest Neighbor Algorithm is NOT an example of an **optimal algorithm** since it does not always find the optimal solution.

Example 2: A courier is based at the head office (labeled A) and must deliver documents to four other offices (labeled B, C, D, E). The estimated travel time between each of these offices is shown on the graph below. The courier wants to visit the locations in an order that takes the least time. Use the nearest neighbor algorithm to find an approximate solution to this problem. Calculate the time required to cover the chosen route.

