- 1. Nearest Neighbor Algorithm: Heuristic algorithm for finding Hamiltonian circuits in complete graphs.
 - 1. Start at a predetermined starting vertex.
 - 2. Identify the nearest unvisited vertex.
 - 3. Go to that nearest unvisited vertex.
 - 4. If there are unvisited vertices remaining, return to step two. Otherwise, return to the starting vertex.

Idea: Make the best choice at each move, based only on the edges you can see from the current position.

- 2. Sorted Edges Algorithm: Heuristic algorithm for finding Hamiltonian circuits in complete graphs.
 - 1. Arrange edges of a complete graph in order of increasing cost.
 - 2. Identify all admissible edges.
 - (An edge is admissible if:
 - i. Selecting it doesn't result in three selected edges meeting at a point.
 - ii. It doesn't complete a circuit that's smaller than the whole graph.)
 - 3. Add the least-cost admissible edge to your circuit.
 - 4. Repeat from 2. until you obtain a Hamiltonian circuit.

Idea: Make the best choice at each move, based on viewing the graph as a whole.

3. Kruskal's Algorithm: Algorithm which finds the minimum weight spanning tree in a connected graph.

- 1. Arrange edges of a graph in order of increasing cost.
- 2. Identify all admissible edges.
 - (An edge is admissible if adding it to your tree does not result in a circuit.)
- 3. Add the least-cost admissible edge to your tree.
- 4. Stop when the tree is a spanning tree.

Idea: Add the smallest edges allowed at each step, much like sorted-edges algorithm with a different idea of admissible.