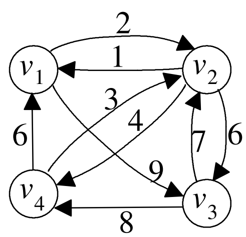
1. *Traveling Salesperson Problem* (TSP) -- Find an optimal (i.e., minimum length) tour when at least one tour exists. A *tour* (or *Hamiltonian circuit*) is a path from a vertex back to itself that passes through each of the other vertices exactly once. (Since a tour visits every vertice, it does not matter where you start, so we will general start at v_1.) What are the length of the following tours?

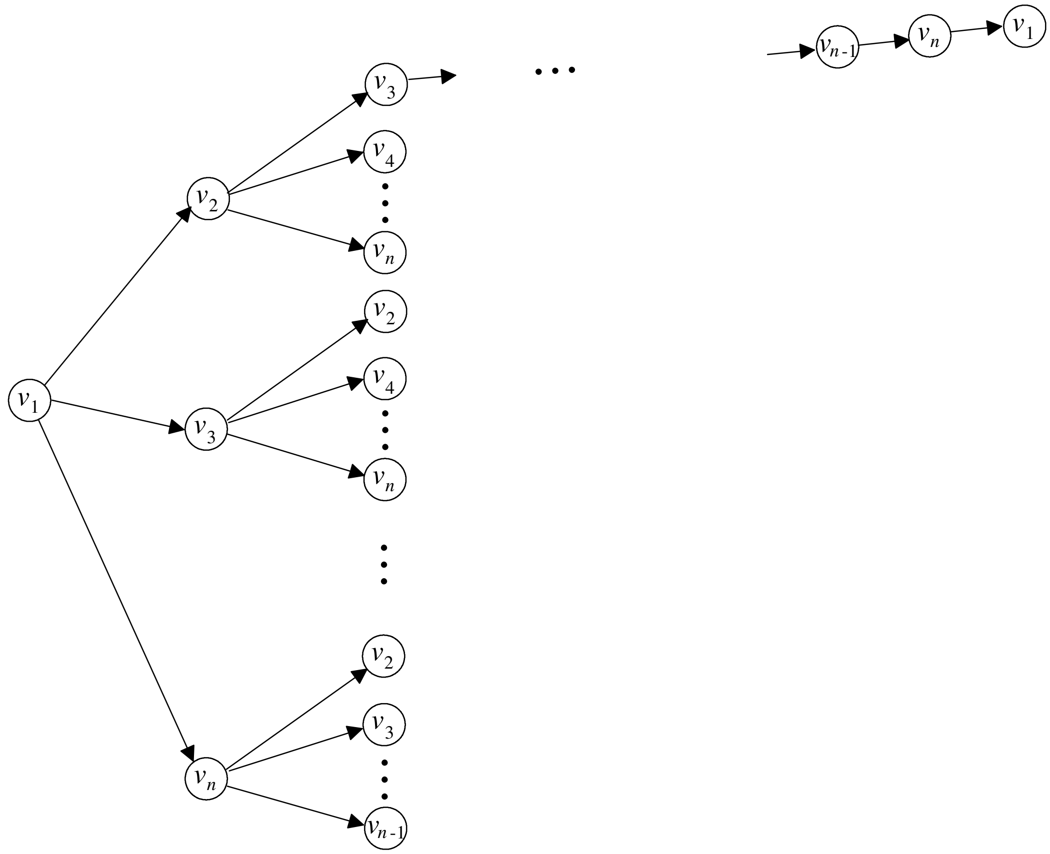


a) $[v_1, v_2, v_3, v_4, v_1]$

b) $[v_1, v_3, v_2, v_4, v_1]$

c) List another tour starting at v_1 and its length.

d) For a graph with "n" vertices (v_1, v_2, ... ,v_n), one possible approach to solving TSP would be to brute-force generate all possible tours to find the minimum length tour. "Complete" the following decision tree to determine the number of possible tours.



The TSP problem belongs to a class of problems known as *NP*-hard problems. Unfortunately, the fastest known algorithms for all *NP*-hard problems are $\Theta(2^n)$. (However, nobody has been able to prove that a polynomial algorithm (e.g., $\Theta(n^2)$, $\Theta(n^3)$, etc.) is not possible. This is one of the biggest open questions in Computer Science: "Does $P = NP$?") If you need to find the solution to an instance of an *NP*-hard problem, what can you do?

- Backtracking (and Branch-and-Bound): Eventhough the worst case is still $\theta(2^n)$ or $\theta(n!)$, these often give efficient results for large problems if we can "prune" **a lot** of branches in the search space tree above.
- Polynomial-time algorithms might exist for a subclass of NP-hard problem, e.g., for TSP with a restricted graph.
- Approximation algorithm - gives good, but not necessarily optimal solutions. Usually, the solution is bounded by how far from optimal its solution is (e.g., approximate solution at least half as good as optimal).

Approximation Algorithm for TSP with Triangular Inequality

Restrictions on the weighted, undirected graph G=(V, E):

1. There is an edge connecting every two distinct vertices.
2. Triangular Inequality: If W(u, v) denotes the weight on the edge connecting vertex u to vertex v, then for every other vertex y,

$$W(u, v) \leq W(u, y) + W(y, v).$$

NOTES:

- These conditions satisfy automatically by a lot of natural graph problems, e.g., cities on a planar map with weights being as-the-crow-flys (Euclidean distances).
- Even with these restrictions, the problem is still NP-hard.

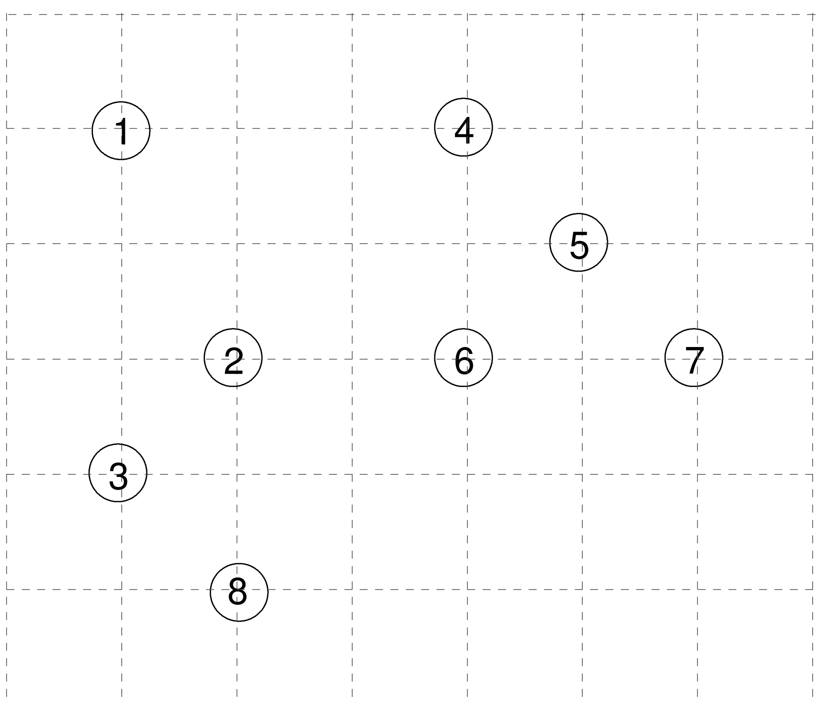
A simple TSP approximation algorithm:

1. Determine a Minimum Spanning Tree (MST) for G (e.g., Prim's Algorithm section 4.1)

2. Construct a path that visits every node by performing a preorder walk of the MST. (A *preorder walk* lists a tree node every time the node is encounter including when it is first visited and "backtracked" through.)

3. Create a tour by removing vertices from the path in step 2 by taking shortcuts.

1. Determine a Minimum Spanning Tree (MST) for G (e.g., Prim's Algorithm) if we start with vertex 1 in the MST. (Assume edges connecting all vertices with their Euclidean distances)

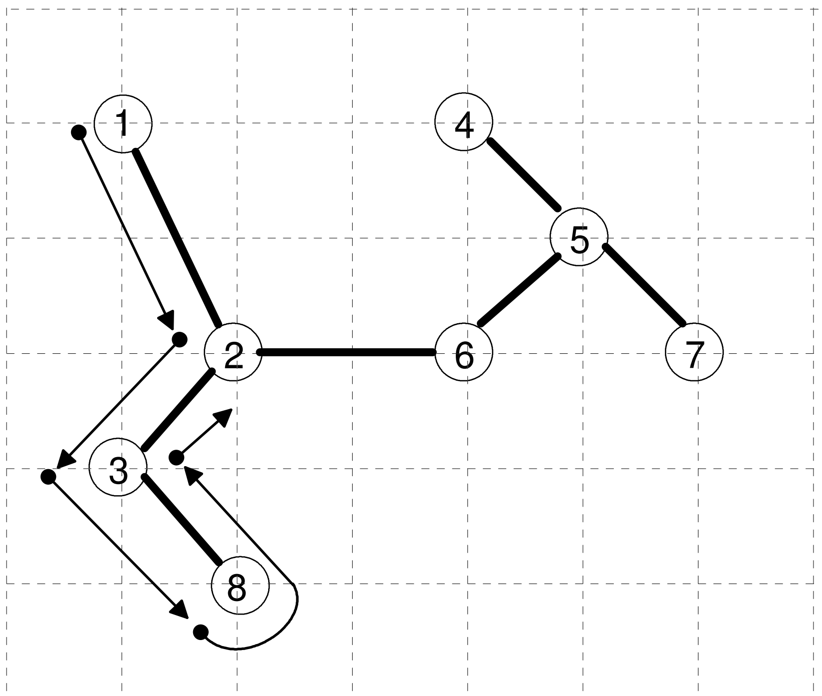


Prim's algorithm is a greedy algorithm that performs the following:

a) Select a vertex at random to be in the MST.

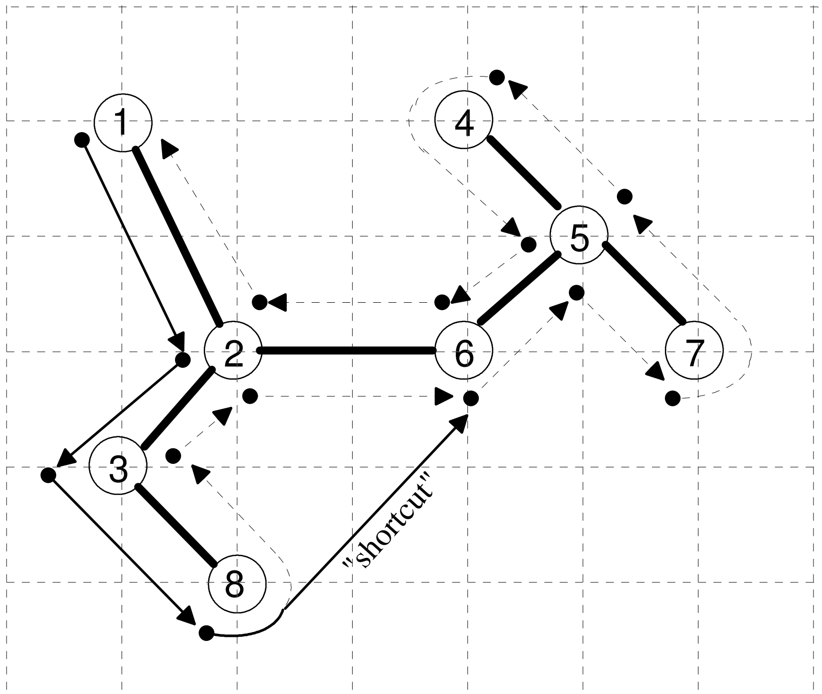
b) Until all the vertices are in the MST:
   - Find the closest vertex not in the MST, i.e., vertex closest to any vertex in the MST
   - Add this vertex using this edge to the MST

2. Complete a path that visits every node by performing a preorder walk of the MST. (A *preorder walk* lists a tree node every time the node is encounter including when it is first visited and "backtracked" through.)



Complete the path constructed by a preorder walk around the MST: [1 2 3 8 3 2

3. Complete a tour by removing vertices from the path in step 2 by taking shortcuts.



a) Finish removing vertices from the preorder-walk path to create a tour by taking shortcuts:
[1 2 3 8 ~~3~~ ~~2~~ 6 5 7 5 4 5 6 2 1]

b) When scanning the above path, how did you know which vertices to eliminate to take a shortcut?

c) If we take the optimal TSP tour and remove an edge, what do we have?

d) What is the relationship between the distance of the MST and the optimal TSP tour?

e) What is the relationship between the distance of the MST and the distance of the preorder-walk of the MST?

f) What is the relationship between the distance of the preorder-walk of the MST and the tour obtained from the preorder-walk of the MST?

g) What is the relationship between the tour obtained from the preorder-walk of the MST and the optimal TSP tour?