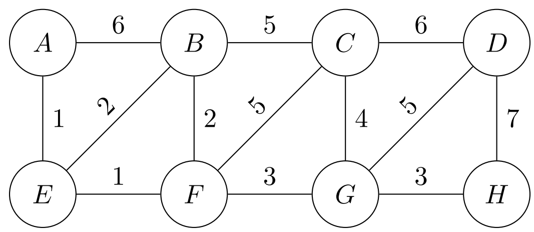
Weekly Assignment 7:
Greedy Algorithms

1. Consider the following graph:



   (a) What is the cost of its minimum spanning tree?
   (b) How many minimum spanning trees does it have?
   (c) Run Kruskal's algorithm on the graph. For each iteration, indicate which edge is added to the spanning tree.

2. Consider a straight road with houses scattered very sparsely along it. You want to place cell phone base stations at certain points along the road, so that every house is within four miles of one of the base stations. Give an efficient algorithm that achieves this goal, using as few base stations as possible. Input for the algorithm is an ordered list $x_1, \ldots, x_n$ of house positions, and the output is a list $y_1, \ldots, y_k$ of positions of the base stations.

3. Nancy wants to throw a party and is deciding whom to call. She has n people to choose from, and she has made up a list of which pairs of these people know each other. She wants to pick as many people as possible, subject to two constraints: at the party, each person should have at least five other people whom they know *and* five other people whom they don't know.

   Give an efficient algorithm that takes as input the list of n people and the list of pairs who know each other and outputs the best choice of party invitees (which could be the empty set). Show the correctness of your algorithm and give the running time in terms of n.

4. Suppose that $G = (V, E)$ a connected graph and $T = (V, A)$ is a spanning tree of G. Prove the following statements or give a counterexample:

   1. If a graph G has more than $|V| - 1$ edges, and there is a unique heaviest edge, then this edge cannot be a part of a minimum spanning tree.
   2. The shortest-path tree computed by Dijkstra's algorithm is necessarily an MST.
   3. If at least one edge of minimum weight of G is in T then T is a MST.

5. Consider a scenario where a group of people P are applying an internship at a set of companies C, where each person and company has a complete preference order of which company or person they prefer.

   By employing the Gale-Shapley algorithm we get a stable matching which has the property the applicant group always get their best stable matching and (somewhat astonishingly) their match cannot be improved by misrepresenting their preferences. Prove or give a counterexample to the following claim:

A company $c \in C$ cannot get a better stable matching by lying about their preference.