1. (10 points) You are given a connected graph G with n vertices and with positive edge weights $w : E \to \mathbf{R}^+$. You wish to find a connected subgraph of G with $n-1$ vertices that has minimum total cost.

   (Please provide an algorithm description, correctness proof and runtime analysis.)

2. (12 points) Suppose you are managing a computer network of n computing sites. Between some pairs of computing sites, there is a link that has a positive initialization time. When you turn on the entire network, all links start initializing at the same time. The whole network is not operational until all links have been initialized. You wish to remove links so that the network stays connected yet you have minimized the maximum initializion time.

   The network is given to you as a connected undirected graph with positive edge weights. You can assume that $|E| = O(|V|)$.

   Design an algorithm that achieves this goal.

   (Please provide an algorithm description, correctness proof and runtime analysis.)

3. (20 points) [4 points for part a, 4 points for each counterexample, 8 points for proving the correct algorithm is optimal]

   You are given two sets of n points each on the number line: $(A[1], \dots, A[n])$ and $(B[1], \dots, B[n])$ (all values are different and each list is sorted in increasing order.)

   You wish to pair up the points (one from the first list and one from the second list):

$$[(A[1], B[i_1]), (A[2], B[i_2]), \dots, (A[n], B[i_n])]$$

   (such that each point is in exactly one pair.)

   You wish to pair them up in such a way to minimize:

$$\sum_{k=1}^{n} |A[k] - B[i_k]|$$

   a) Describe this problem as we have done in class in terms of:
      - **Input:**
      - **Solution Format:**
      - **Constraints:**
      - **Objective:**

   b) **Candidate Greedy Strategy I:** Find the pair $A[i], B[j]$ that is the closest (with the smallest overall $|A[i] - B[j]|$ distance) and pair $(A[i], B[j])$ (break ties by choosing the smaller value of $A[i]$.) Remove $A[i]$ from the first list and remove $B[j]$ from the second list and repeat on the remaining points until all points are paired.

      Either prove that this strategy always yields an optimal solution or give a counterexample to show that it is not always optimal.

   c) **Candidate Greedy Strategy II:**

      Pair up $(A[1], B[1]), (A[2], B[2]), \dots, (A[n], B[n])$.

      Either prove that this strategy always yields an optimal solution or give a counterexample to show that it is not always optimal.

   d) **Candidate Greedy Strategy III:** Let $B[j]$ be the closest point to $A[1]$ in the B list (break ties by choosing the lower B value). Pair up $(A[1], B[j])$ and remove $A[1]$ and $B[j]$ from the lists and continue with $A[2]$ until all points are paired.

      Either prove that this strategy always yields an optimal solution or give a counterexample to show that it is not always optimal.

4. (16 points) [4 points for part a, 4 points for counterexample, 8 points for proving the correct algorithm is optimal]

   Suppose you are driving along a road in an electric car. The battery of the electric car can bring you $x[0]$ miles. There are battery stations along the way at positive positions $D[1], \ldots D[n]$ (in sorted order.) Each battery station can *replace* your battery and give you a new battery that can bring you a certain number of miles. The distances of the batteries are given in the array $x[0], x[1], \ldots, x[n]$

   You wish to start at position 0 with a full battery and end at position $D[n]$ replacing the fewest batteries.

   a) Describe this problem as we have done in class in terms of:

      - **Input:**
      - **Solution Format:**
      - **Constraints:**
      - **Objective:**

   b) **Candidate Greedy Strategy I:** Travel to the farthest battery station without exceeding $x[0]$ miles. Replace the battery at that station and repeat the process starting from that station until you can reach position $D[n]$.

      Either prove that this strategy always yields an optimal solution or give a counterexample to show that it is not always optimal.

   c) **Candidate Greedy Strategy II:**

      Travel to the battery station with the largest $x[i]$ value without exceeding $x[0]$ miles. Replace the battery at that station and repeat the process starting from that station until you can reach position $D[n]$ and then go directly there.

      Either prove that this strategy always yields an optimal solution or give a counterexample to show that it is not always optimal.

   d) **Candidate Greedy Strategy III:**

      Travel to the battery station with the largest $D[i] + x[i]$ value (in other words, the battery that can take you the farthest down the road.) Replace the battery at that station and repeat the process starting from that station until you can reach position $D[n]$ and then go directly there.

      Either prove that this strategy always yields an optimal solution or give a counterexample to show that it is not always optimal.