1. **Optimizing Cell Tower Placement**: There is a long straight country road with expensive houses scattered along it. The houses are owned by affluent stock traders who require cell phone service. Your company needs to provide the cell service to every house without exception. The towers only have a range of four miles. You want to place the cell phone towers at locations along the road so that no house is more than four miles from the nearest cell phone tower. You are given the exact mileage along the road where each house is located and these numbers are given in increasing order of distance from the start of the road. Design a greedy algorithm that will determine the set of locations for the cell towers that requires the fewest cell tower. **Denote the location of the i^{th} house as M[i], i=1..n]** and the location of the k-th cell tower as LOC[k]
   Give an efficient algorithm.
   a. Specify (pseudo code) an efficient greedy algorithm to achieve this goal with the fewest cell towers.
   b. Prove your algorithm always finds an optimal solution.
   c. What is your algorithm's complexity? Justify your answer.

2. There is a long, straight pipeline with critical monitoring devices attached to it that you want to be able to be able to repair in an emergency. In order to be able to repair them there needs to be an access road that runs up to the pipeline and is within 2 miles (comfortable walking distance) along the pipeline of each monitoring device. Suppose the monitoring devices are at integer distances m(1), …,m(n) from the beginning of the pipeline. **Assume that m(i) < m(i+1) for all i from 1 to n-1**. You want to determine the number and locations of the access roads that you have to build and maintain. Since the access roads are very expensive you want to build the **smallest number** of access roads possible.
   a. Specify (pseudo code) an efficient greedy algorithm to find the smallest number of roads and their locations.
   b. Prove your algorithm always finds an optimal solution.
   c. What is your algorithm's complexity? Justify your answer.

3. **Backpacking planning:** You and some friends are planning a backpacking trip to the Sierras. You want to hike as far as possible each day but do not want to hike at night. On a map you have identified a large set of good camping sites. You plan to use the following system to decide where to camp each day. Each time you come to a one of the identified camp sites, you determine whether you can make to the next good site before nightfall. If you can make it to the next site before nightfall, you will continue hiking otherwise you will stop.
   Although there are drawbacks, e.g. you may miss staying in some great spots, you believe the system has one good feature, namely: "Given that we are only hiking in the day, it minimizes the number of camping stops we will have to make."
   You want to determine if this claim is true or false. To model this problem, you decide to make the following assumptions. Model the trail as a long straight path of length L and assume you and your friends can hike a maximum of d miles each day. Also you check the maps and determine that the campsites are located at distances $x_1, x_2, x_3, \ldots, x_n$ along the trail, with $x_i < x_{i+1}$ for all i. Also, you believe that group will always correctly determine whether or not they can reach the next campsite by nightfall.
   This means we can consider a set of stopping points **valid** if the distance between each adjacent pair is ≤ d. Thus, if you stop only at the points (campsites) in a **valid** set you will be able to make the full trip in daylight. You know it is possible to make the full trip in daylight since the full set of campsites is valid.
   a. Specify, in pseudo code, an efficient greedy algorithm that determines at which campsites the group will stop.
   b. Prove your algorithm always finds a smallest set of valid campsites.
   c. What is your algorithm's complexity? Justify your answer.