1. Alan has just graduated with a degree in Computer Science and Engineering, and being unable to find any other job has joined Willy Wonka's chocolate factory for delivering chocolates from his factory in Plymouth to Aberdeen. Alan gets to drive the battery-powered Chocolate Delivery Vehicle along Willy's Chocolate Delivery Network. The two cities are connected by a single line of the Chocolate Delivery Network.

   The Chocolate Delivery Vehicle runs on single-use batteries that must be replaced after at most 100 miles. There are specific Battery Replacement Stations within the direct line from Plymouth where the batteries can be replaced. Alan uses Google Maps to find the distances of the Battery Replacement Stations from Plymouth, and wants to compute the minimum number of stops that he has to do while transporting chocolates from Plymouth to Aberdeen.

   (a) Alan is given an array $D[1, 2, \ldots, n]$ where $D[i]$ is distance of the i^{th} Battery Replacement Station from Plymouth. Help Alan design a greedy algorithm to compute the minimum number of stops that must be made to replace batteries. He is adamant that he will only accept provably correct solutions. It is also known that $D[i] - D[i-1] \leq 100$ so that the vehicle never runs out of fuel.

   As Alan is about to set out, he realizes that each Battery Replacement Station charges a different amount for replacing the battery. Furthermore, even if you have some battery left, you still have to pay the entire amount for replacing it with a new battery. Together with the array $D[1, 2, \ldots, n]$, Alan also obtains another array $C[1, 2, \ldots, n]$ where $C[i]$ is the cost of replacing the battery at the i^{th} station. Alan realizes that it is not enough to find the minimum number of stops, but rather he has to find the stops that minimizes the total cost of replacing batteries.

   (b) Convince Alan that the greedy algorithm in Part (a) is not sufficient to find the number of stops that minimizes the cost by giving a suitable counter-example.

2. Given an integer n, consider the following process to obtain n starting from 1 using the following operations: you are allowed to increment the current number by 1, or double the number. Give an efficient greedy algorithm to find the minimum number of such operations to obtain n from 1.

3. Let $\mathcal{I}$ be a set of intervals on the real line. A subset $\mathcal{I}' \subseteq \mathcal{I}$ covers $\mathcal{I}$ if the union of interval in $\mathcal{I}'$ is equal to the union of intervals in $\mathcal{I}$. The size of the cover in the number of intervals in $\mathcal{I}'$. Describe a greedy algorithm to compute the cover of smallest size for an input $\mathcal{I}$ of intervals. You can assume that the intervals are input by giving their left and right endpoints.

4. Let $\mathcal{I}$ be a set of intervals on the real line. A set of points P is said to *stab* $\mathcal{I}$ if every interval of $\mathcal{I}$ contains at least one point of P. Describe a greedy algorithm to compute the smallest set of points P that stabs $\mathcal{I}$. You can assume that the intervals are input by giving their left and right endpoints.

5. Karl wants to organize another party. After the last party turned out to be a dud, Karl has decided to make sure that every person in the party knows at least three others, and also does not know three others (so that the party is more lively). For each of Karl's friends, Karl knows who is friends with whom. Give an efficient greedy algorithm to find the largest set of friends that Karl can invite satisfying this constraint. Assume that the friendship relation is symmetric, and that the largest set of friends satisfying these constraints could also be empty.

6. Let $\mathcal{I}$ be a set of intervals on the real line. A *coloring* of $\mathcal{I}$ with k colors is a function $C : \mathcal{I} \to \{1, 2, \ldots, k\}$ such that if $I, I' \in \mathcal{I}$ overlap, then $C(I) \neq C(I')$. Describe a greedy algorithm to compute the smallest k, and a coloring $C : \mathcal{I} \to \{1, 2, \ldots, k\}$. You can assume that the intervals are input by giving their left and right endpoints.

7. Suppose you are given two arrays $R[1, 2, \ldots, n]$ and $C[1, 2, \ldots, n]$ of positive integers. A $n \times n$ binary matrix M is said to agree with R and C, if the i^{th} row of M contains $R[i]$ many ones, and if the i^{th} column of M contains $C[i]$ many ones, for every i. Given R and C, give an efficient greedy algorithm to check if such an M exists? If so, construct a matrix M that agrees with R and C.