CS2800 Design and Analysis of Algorithms
Problem Set #6 March 6, 2023

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.

   Solution: Choose the Battery Replacement Station that is closest to the 100 mile mark. Take any other optimum solution. We can replace the first station by the one closest to the 100 mile mark and it will still be a valid solution. Keep modifying inductively.

   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.

   Solution: Let $n = 2$, and $D[1] = 50$ and $D[2] = 98$ and the destination is at distance 140. Now, $C[1] = 20$ and $C[2] = 100$. Choosing Station 1 gives a cost 20, whereas the greedy choice will choose Station 2 and this has a cost of 100.

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.

Solution: We will look at starting from n and either decrementing by 1 or halving n.

Starting from n make the following choices.

- If n is odd, decrement n by 1.
- If n is even, halve n.

For odd n, the only option for the first step is to decrement n by 1.

Suppose that for even $n = 2k$, the first step in the optimal solution is decrementing by 1. Let the first halving step in the optimal solution be after $2l$ decrements. Thus, the optimal solution includes $2l$ decrements and then a halving which gives $k - l$. Replace this part of the solution by halving n (this gives k) followed by l decrements. This has lesser number of steps and contradicts the optimality of the original solution.

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.

Solution:

Greedy choice: Let p be the first point that is not covered. Find the interval that covers p and extends the longest. Add this to $\mathcal{I}'$ and continue.

Use a similar argument as in the previous problem where you try to change an optimal solution by removing an interval and adding the interval given by the greedy choice.

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.

Solution:

Greedy choice: Find the interval I that finishes first, and add its finishing point to the set P. Remove I and all the intervals overlapping with I, and recurse.

Every interval contains at least one point from P. In particular the interval that finishes first as well. Now if there is a point in P that lies in this interval, remove this point and add the finishing point. The size of set cannot increase, and the new point will stab all the points that were stabbed by the earlier point since no interval finishes before this interval (the greedy choice chose the interval that finished first).

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.

Solution: Construct an undirected graph with the friends of Karl as vertices and the edge relation same as the friend relation. The goal is to find a largest set of vertices V' such that in the induced graph $G[V']$ every vertex has degree ≥ 3, and the complement graph also has degree ≥ 3.

Greedy choice: Remove a vertex v in G that does not have this property together with its incident edges. The vertex v can never be part of any solution. Keep doing this, and whatever remains is the solution.

6. Let $\mathcal{I}$ be a set of intervals on the real line. A *coloring* of $\mathcal{I}$ with k colors is a function $C : \mathcal{I} \rightarrow \{1, 2, \ldots, k\}$ 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 $C : \mathcal{I} \rightarrow \{1, 2, \ldots, k\}$. You can assume that the intervals are input by giving their left and right endpoints.

Solution:

Greedy choice: Starting from the interval with the earliest starting time, assign the smallest unassigned color.

If there is a set of k intervals that all intersect with each other, then you cannot color with less than k colors. To prove optimality, we want to show that the greedy choices will use no more than k colors.

Suppose that while we consider an interval I with start time $s(I)$ and end time $f(I)$, we are forced to use color $k+1$. This means that there are k intervals that are colored that overlap with this interval. Since we choose intervals in the increasing order of starting times, it must be the case that the start time of all these intervals is at most $s(I)$. Furthermore since they overlap with I, their end times is at least $s(I)$. Therefore, the point $s(I)$ *stabs* all these intervals and hence all these intervals must intersect amongst themselves also. But, this would mean that there are $k+1$ intervals that all overlap with each other, which contradicts the assumption that there are at most k intervals that all overlap with each other.

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.

Solution: Firstly, $\sum_{i=1}^{n} R[i]$ must be equal to $\sum_{i=1}^{n} C[i]$ for there to be a solution.

Greedy choice: For each i, do the following:

1. If $|\{j \mid C[j] > 0\}| < R[i]$, say no solution exists.
2. Find the $R[i]$ largest values $C[j]$, and set those columns in row i to 1.
3. Reduce $C[j]$ by 1 for all the columns that was set to 1 in the previous step.

Suppose that there is a matrix M that agrees with R and C. Consider the first row. Suppose that $M[1, j] = 0$ and $M[1, j'] = 1$ and $C[j] > C[j']$. Therefore, there exists an i' such that $M[i', j] = 1$ and $M[i', j'] = 0$ since otherwise it will contradict the fact that $C[j] > C[j']$. So, if we swap the values $M[1, j] \leftrightarrow M[i', j]$ and $M[1, j'] \leftrightarrow M[i', j']$, the row sums of 1^{st} and i^{th} rows are unchanged and the columns with larger $C[j]$ values are set before the columns with smaller $C[j]$ values just like in the greedy choice.

Show that this idea works at each step of the algorithm.