CPSC 320 202W1: Greedy Tutorial Problems

Night at the Museum

Suppose that you're in charge of hiring security guards to protect priceless artifacts in a museum exhibit. You are given a line L that represents a long hallway in the show room. You are also given an unordered set $X = \{x_0, x_1, \ldots, x_{n-1}\}$ of real numbers that represent the positions of artifacts in this hallway. Suppose that a single guard can protect all the objects within distance at most d of his or her position, on both sides.

1. Design a greedy algorithm for finding a placement of guards that uses the minimum number of guards to guard all the artifacts with positions in X.

   The basic idea behind our algorithm is that, since a guard needs to cover the artifact at the leftmost position, we might as well place them as far to the right as possible, while still covering that leftmost object (that way, we increase the chances that they'll be able to cover other artifacts as well).

   In terms of our actual algorithm: we begin by sorting the artifacts from left to right. We then place a guard distance d to the right of the first artifact in the list, and remove from the list all artifacts that are within d of the guard's position (because now they're covered). We then repeat with the rest of the list and continue until all artifacts are covered (i.e., when the list is empty).

2. Analyze the running time of your algorithm as a function of n, the number of objects that need guarding.

   The sorting step takes $O(n \log n)$, and the remainder of the work is linear (as it can be essentially done by iterating once through the list). Therefore, the algorithm has time complexity $O(n \log n)$.

3. Give a "greedy stays ahead" lemma that you could use to prove your approach is optimal. You do not need to prove the lemma here (though you will need to do this in the next part of the question).

   Consider the guards in the greedy solution, ordered from left to right, and the guards in the optimal solution, ordered from left to right. Our stays-ahead lemma is: the ith guard in the greedy solution is at least as far to the right as the ith guard in the optimal solution.

4. Prove that your algorithm is optimal – i.e., that it guards all artifacts and does so using the fewest possible guards.

   Before we proceed, we need to show that the greedy solution is in fact valid (i.e., leaves no unguarded artifacts): this is easy to deduce from our algorithm because, with each guard we place, we only remove from the list those artifacts that are protected by the guard (i.e., that are within d of the guard's position). The algorithm only terminates when the list is empty (all artifacts are guarded) so we conclude that the greedy solution does protect all the artifacts.

   Now we'll prove our stays-ahead lemma by induction on i. Clearly, the first guard in the optimal solution can be no further to the right than the first guard in the greedy solution: in the greedy algorithm, we placed the guard as far to the right as we could without leaving the leftmost artifact unguarded. Therefore, the optimal solution can't place the a guard further to the right.

Now, assume that guard $(i-1)$ in the greedy solution is at least as far to the right as guard $(i-1)$ in the optimal solution. The first object unguarded by guard $(i-1)$ (and all previous guards) in the greedy solution is therefore **at least as far to the right** as the first object unguarded by guard $(i-1)$ (and all previous guards) in the optimal solution. So, the optimal solution can't place guard i further to the right than the greedy algorithm places guard i: again, the greedy solution places guard i as far right as possible while still guarding the first artifact not guarded by guard $(i-1)$, and the optimal solution also has to guard that same artifact (and potentially some other artifacts further to the left). This means it can't place guard i any later.

So, we've shown that the ith guard in the greedy solution is at least as far to the right as the ith guard in the optimal solution. We also established that the greedy solution protects all the artifacts. The optimal solution cannot have fewer guards than the greedy solution: this would mean that it had placed a guard far enough to guard the rightmost object **earlier** than the greedy solution did, which contradicts the result we showed earlier that the greedy solution's ith guard is at least as far to the right as the optimal solution's ith guard. Because our definition of "optimal" was to protect all the artifacts using the fewest possible guards, we conclude that our greedy algorithm is in fact optimal.