Homework 3 Writeup

Jackson Hart

April 29th, 2022

Problem 1

A)

The algorithm I made finds the furthest away hotel that is within d miles, and selects that one. It does this by first sorting by how far away the hotels are in descending order, and then goes down the line checking if they're within d miles of the current destination and the first one to meet this criterion is selected.

Algorithm 1 Road Trip

```
Require: x is a list containing the distance of the ith hotel from the start,
   and contains the property x[i].index which is the index of that hotel.
Require: Sort(x) sorts x in descending order.
   Sort(x)
   hotels ← empty list
   for $i \leftarrow 0$ to $n$ do
       while $x[i] > d$ and $i \neq n$ do
           $i \leftarrow i + 1$
       hotels.append(x[i].index)
```

B)

The theoretical running time of this algorithm is $O(n)$ and $\Omega(1)$, giving $\Theta(n)$.

Problem 2

We already know that this problem has optimal substructure from the readings and lecture, so that part is proven. In this case, the greedy choice is always choosing the latest starting activity because in each iteration, the latest starting activity is a locally optimal solution, and it always returns a globally optimal solution.

Proof. Consider the set of activities A such that $A = \{a_1, ..., a_n\}$. If $n = 1$, then we are done. If $n = 2$, then we will select the latest activity to start. If the first activity is compatible then we have the optimal solution, if it isn't, then if we had selected the first activity we would have the same number of activities and thus we have an optimal solution. Now, let n be some arbitrary number such that $n \geq 1$. Consider some optimal solution, S, to A that does not contain the latest starting activity, a_i. If a_i is compatible, then we have a contradiction and now have the optimal solution, if it is not, then if we remove the latest starting activity in S, s_i, we can replace it with a_i because the starting time of $a_i \geq s_i$. Because S is the same size, we now have another optimal solution, and therefore, there will always be an optimal solution with the latest starting activity. □