

CS 325 - Homework 3

Alexis Doyle

#1 (a) Create a for loop for the day since you may only drive during the day. A while loop would also work, while $! = \text{destination}$.

In this loop we can put another loop to cycle through and check hotel options. This loop will be checking the distance of the hotels and if the distance is $>$ than distance of the previously stayed hotel. If the distance is greater then you select this hotel.

The loop will continue until the destination is reached.

(b) the worse case is there are multiple options for hotels distance which means we may have to calculate the distance of the hotels multiple times giving a complexity of $O(2n) = O(n)$.

```
While (Place != destination) {  
    for ( i < # of hotels ) {  
        if ( hotel distance > 'd' last hotel ) {  
            // Pick this hotel  
            break;  
        }  
    }  
}
```

#2 If we instead select the last activity to start with we are essentially doing the same thing as before but backwards/reverse. So it is really the same problem and can be solved using a greedy algorithm because we can pick the best looking option at every step.

Proof explanation:

Let A_k be the max size of a subset of mutually compatible activities in S . Let a_i be an activity in the subset A_k with the last start time. If a_m is the max size of a subset and $a_m = a_i$ then we are done with the last to first. If $a_m \neq a_i$ then we are not done with the last to first algorithm. So let $A_k' = A_k - \{a_i\} \cup \{a_m\}$ and $|A_k'| = |A_k|$.

#3 Pseudocode:

```
int main () {
```

```
    while not end of file {
```

- load in data for start[] + finish[]
- Bubble sort both arrays $\rightarrow O(n)$ runtime
- run arrays through last to start algo.
- Print results

```
    }
```

```
}
```