Joaquin Saldana
CS325 / Intro to Algorithms
Homework 4

Problem 1: **Class Scheduling**

In order to do this efficiently, you will need to maintain two lists of lecture halls.

1. Halls that are busy at start time, st
2. free at the finish time, ft.

The algorithm will require sorting the classes.

When you need to schedule a class, move the hall from the free list to the busy list. Same goes for when a busy hall is no longer in use, you move it to the free list. You initialize with zero halls. If there are no halls in the halls free list then you will need to create one.

The movement of the halls in between the 2 lists/arrays is constant time. The total running time is dominated by the sorting component which is O(n lg n)

Problem 2: **Road Trip**

I'm envisioning an array, A[h0, h1, h2, …. , hn] w/ hn as the final hotel, with the hotels sorted from shortest distance to longest distance. Each day, you iterate through the array, picking up from the hotel you just stayed at, and compare the distance to the distance your're allowed per. You keep iterating through the array until you've reached a hotel distance > than your allowed distance, hj, and you select the hotel that is hj-1. You keep repeating until you reach hn, your destination.

The running time should be linear, O(n), with at worst being O(2n) when you need to need distances more than once.

Problem 3: **Scheduling jobs w/ penalties**

$We\ can\ assign\ time\ intervals\ for\ the\ minutes, M_I\ , for\ i\ is\ 1 \leq i \leq n.\ \ We\ then\ do\ the\ following$:

- Sort the jobs in decreasing order of the penalties
- To add a job, if a time interval is available, we then schedule it in the last such available interval. Else, we use the first available interval starting backwards from M_n

The run time for this algorithm would be O(n^2) if we search as far as long as the last possible time.