HW3

1. Road Trip
   1. I would design an algorithm that looked at your current position and the most miles you could drive in a day. It would add the two and find the closest hotel on the near side of d, the maximum driving distance. It would then repeat the process until you reached your destination

Psuedocode:

hotels = n

X = [x1,x2,x3,x4,…,xn]

d = max drive distance

pos = 0 //distance in miles

numhotel = 0 //number of hotel that you’re at 1st 2nd 3rd

days = 0

while(pos<X[n-1]) ­­//while we haven’t reached the destination

for i=numhotel; i<n

if X[i] > pos+d //check through all the hotels, starting with the current one to see if we can drive to them

if i == numhotel

break //the next stop is too far away trip is impossible

else

pos = X[i-1] //update where you are, what hotel you’re at, and the days

numhotel = i+1

days ++

return days

1. Last-to-start Greedy

This approach is still a greedy algorithm because it maximizes the time before the current activity. Assuming the algorithm is run ahead of time, activities can be scheduled start-to-finish or finish-to-start, and the same number of activities can still be run.

To prove this algorithm works, consider the case where the “optimal” solution contains an activity that isn’t the last to start. Then by definition, there is an activity with a later start time, that could either be substituted in for the non-last-to-start activity, or added onto the end of the solution set, depending on its size. The former case would yield the same optimal solution, and the latter would yield the true optimal solution. In both cases the optimal solution contains the available activities with the latest start times.

1. Last-to-start code

--In files