Hw6

Question2:

Algorithm idea:

The basic idea is greedy algorithm. Assume we know the distance between all gas station and buffalo, so we can use merge sort to get an ordered list (From close to far) as input. Then loop through all the gas station if that distance is less than the deadline, we set the possible stop to this gas station until we reach the first gas station which distance greater than the deadline. We now can put that into the actual list of stop. And then set the next deadline by adding 350 to the current gas station’s distance and also the next possible stop to be the first gas station that has greater distance to previous deadline. The initial position is 0, and we just need to keep updating the current gas station’s distance and plus 350, then we get the max range the driver can reach which is the next deadline and looking for the most far gas station within the range. So we found next bus station. Therefore, Driver can keep following this idea until she reach the Seattle.

Algorithm detail:

Assume we have a list of distance gas station from Buffalo (unorderedList) as input

distanceList= MergeSort(unorderedList) #O(nlog(n)) get a orderedlist

Deadline = 350 #set the first deadline to be 350 miles

Answer = [ ] # answer will be list of the index of stop

gasStationCount = 0 #counter value to count the index of the stop

For distance in distanceList: #go through all the gas station

If distance <=Deadline: # if the current gas station is still

# within the deadline

#set the possibleStop to the current distance

possibleStop = distance

else: # else add the previous possible Stop into

# the actual answerlist

Answer.append(gasStationCount-1)

Deadline=possibleStop +350 # set the new deadline by adding 350 to

#current

possibleStop = distance # set the new possible Stop to the

# current distance

gasStationCount++ #increment the gasStation counter

endFor

return Answer

Proof of Correctness idea:

I assume her trip to Seattle is a straight line and in between these two points there are several gas station. And because it is a straight line I just label the gas station from close to far with their index numbers respectively. Since I use mergesort to obtain a sorted list, my idea is to basically find a gas station as far as possible from the current position before reach the respective deadlines until the gas is enough for her to reach Seattle. Therefore, my algorithm can find the minimum number of gas stations she need to stop.

Proof detail:

As I state in Proof idea, her trip can be view as from one point to another point and in between two points, there are several points for her to restore gas which mean make a new initial position and search for next gas station as far as possible within the range 350 from current position. As given, I have all the distance from each of the gas station to Buffalo, and it is a preordered list which from closest to the farthest gas station as my input. My algorithm go through all of the gas stations just as its order and make them all as a possible stop which Ms.LiberalElite going to stop. And then as I found a gas station that satisfy the farthest distance within the range. I will then update the deadline and keep going with the remaining gas station until I reach the end. Therefore, there is no way I can miss a gas station that is within the range. Since as the question instruction given, I can assume there are no two gas stations are more than 349 miles apart. So as follow my algorithm, there should be no way I can miss a deadline since my search range is always from a gas station to 350 miles forward. As a result, as my algorithm run to the end, I can get the minimum number of gas station that Ms.LiberalElite need to stop in order to reach Seattle.

Runtime Analysis:

MergeSort: O(nlog(n)) times in order to get my ordered list

Initialize Deadline, answerlist and gasStationCounter : Constant time O(1) time for each

For loop: O(n) since it loop through all of the gas stations.

Overall: O(n) times