CS 5633: Analysis of Algorithms

# Homework 7

Shimul Debnath(sye680)

1. a) In this case, the algorithm will sort the array based on the smallest active time.

So, if the activity sequence is like (0, 5), (6, 10), (4, 7), (10, 20)

This algorithm will select activities: (4,7) and (10, 20) only while the optimal solution is (0,5), (6,10), and (10, 20).

So, here least amount of active time algorithm fails.

b) Algorithm: Activities based on the last start time will return an optimal solution.

We have to sort the activities based on decreasing start time. So, the first pair will be the last started activity.

A = {a1}

ak = a1

for i = 2 to the total no of activities

if finish time of ai <= start time of ak

ak = ai

A = A U {ak}

Proof of Correctness:

Consider any non-empty subproblem Sk and let ak denote the activity in Sk with the last start time. Then ak is in some optimal solution of Sk.

Proof:

Let Ak be an optimal solution for Sk and let ai be the activity in Ak with last start time. If ak = ai, we are done. So, suppose ak != ai. Consider Ak” = (Ak U {ak}) \ {ai}

Activities of Ak have a finish time earlier than the start time of ai and ak has a start time greater than the finish time of all the activities of Ak. So, |Ak”| = |Ak|.

So, Ak” is optimal and hence proved that provided algorithm is correct.

1. We have to place a tower within 4km of each house. So, let distance be an array that has the distance of each house from the start of the road.

func place\_tower(distance)

total\_tower = 0

tower\_loc = 0

for i = 1 to n:

if no tower within 4km left and 4km right

tower\_loc = distance[i]+4

total\_tower += 1

return total\_tower

Proof of Correctness:

This algorithm checks whether there is a tower within 4 km both on the left and right side. If there is a tower within this distance, no need to place an additional tower. Otherwise, a tower will be placed.

Let Ak” is the optimal solution that has fewer towers than solution Ak provided in the above algorithm. So, there must be a house in the provided algorithm which has placed at least two towers within 4km left and 4km right of one house. But this is not possible with the provided algorithm. So, it is not possible to find a solution Ak” which has fewer towers than Ak. Hence, |Ak”| = |Ak|. So, provided algorithm is correct.

Running time:

The running time = O(n)