Davie Truong

I have read and agree to the collaboration policy. Davie Truong

Homework Heavy

CMPS 102 – Spring 2017 – Homework 3

Solution to problem 2

A.

i. Suppose we go with the proposed algorithm. If we choose the power plant with the smallest index that fits the condition of a five-mile distance, then we neglect the second variable which is , the quality of electricity. If there exist a power plant n+1, that is adjacent to the algorithms proposed location, fits within the constraints, and has a higher , then it would be more optimal to choose that n+1 power plant.

ii. This algorithm aims to choose the most profitable power plant in the sorted list of where it is sorted highest to lowest based on energy output, i.e. Greedy on . This algorithm will not work because when it chooses the highest , it sacrifices the compatibility aspect in which it doesn’t choose the best output of total sums. Ex: (2, 8), (5, 7), (10, 9), (12, 13), (15,12). This algorithm would choose (12, 13) and (2, 8) resulting in 21 for the total energy. Here as (2, 8), (10, 9), (15, 12) are compatible and have higher total energy.

B. OPT(r) = value of optimal solution to the problem consisting of maximum total energy.

i. Case 1: OPT selects power plant location .

- Can’t use incompatible location

- must include optimal solution to problem consisting of remaining compatible positions with

high energy

Case 2: OPT does not select power plant location

- must include optimal solution to problem consisting of remaining compatible positions with

high energy

ii.

OPT® = {0 if = 0 // no locations

{max ( + M-Compt-Opt (), Compt-Opt(I – 1)) // is compatible positions

This formula is correct because it recursively adds the energy of each power plant that is compatible, while keeping track of the highest value. This goes through every scenario thus returns the most optimal total energy production.

iii.

input: n, pairs (

compute P (1), P (2), …, P(n)

for I = 1 to n

m[i] empty

m[0] = energy of first position

m-compt-opt-(i)

if (m[i] is empty)

m[i] = max ( + M-Compt-Opt (), Compt-Opt (I – 1))

return m[i]

Proof of Correctness: This algorithm is correct because it utilizes the formula to find the highest summed value and stores it in an array. Thus, if we want to find results to each sub problem we can look for it.

Time Complexity:

Since the algorithm must at most go through n locations and compare it’s max value to n-1 locations, the time complexity is O (

Space Complexity:

O (n) because we create an array of size n to store the max value of after each recursive call.

iv.

Find-Solution(i)

If (I = 0)

Output nothing

Else if (

Print i

Find-solution (P(i))

Else

Find-Solution (i-1)

This algorithm is correct because it recursively steps through the memorized data to find the positions of the highest energy production.

Time Complexity: O (n) because it only needs to step through the array data that has already been processed, thus worse case it would need to return all points.

Space Complexity: O(n) much like time complexity, worse case all the points were compatible so it would need a size n array to return the data.