Parth Ray

Lab 4-1: Greedy Algorithms: Stay Ahead

1. Optimizing Cell Tower Placement:
   1. Place a tower 4 miles after the first house. Then keep placing towers 4 miles after a house if the house is not already within range of a tower.

OptimizingCellTower(M[1…n])

LOC[1] = M[1] + 4;

k = 2;

For i from 2 to n:

If M[i] is not within the bounds of LOC[k - 1] then

LOC[k] = M[i] + 4

k++

Return LOC;

* 1. Proof of correctness: Placing a cell tower 4 miles after a house with no service produces the minimum number of cell towers with full coverage.

Setup: Assume LOC\*[k1…kn] is the optimal solution to the tower placement problem with n number of towers being placed for coverage of each house and let LOC[k1…km] be the set of tower locations given by our greedy algorithm.

Proposition: Let P(k) be the proposition LOC\*[k] ≤ LOC[k]. Want to show that

this is true for all k1 up to n.

Base Case: Show P(1) is true. LOC\*[1] ≤ LOC[1].

Proof of Base Case: LOC\*[1] ≤ LOC[1] is true since the LOC[1] is placed at the

max range away from M[1], thus LOC\*[1] must be less than

or equal to LOC[1] in order to provide service to the first

house.

Inductive Case: Show that P(k) is true → P(k + 1) is true for any 1≤k<m. Thus

we need to show that LOC\*[k] ≤ LOC[k] → LOC\*[k + 1] ≤ LOC[k + 1].

Proof of Inductive Case:

LOC[k + 1] is the tower location that is 4 miles after the location of the

next house that is not covered.

Claim: LOC[k + 1] is placed as far down or further down the road than

LOC\*[k + 1].

1. LOC[k + 1] ≥ LOC[k] since we are placing the towers in increasing

order.

2. LOC[k] ≥ LOC\*[k] by the inductive hypothesis

3. LOC[k + 1] ≥ LOC\*[k]

4. LOC\*[k + 1] ≥ LOC\*[k] in order for it to be in the optimal solution,

you cannot have overlapping towers.

Since, LOC[k + 1] and LOC\*[k + 1] are both ≥ LOC\*[k] and LOC[k + 1] is placed as far down the road possible, in order to cover the house before it, LOC[k + 1] ≥ LOC\*[k + 1]. Thus by Principle of Mathematical Induction, P(k) is true for k = 1..m.

Final Step: Show that m = n. Suppose that m > n. But this is impossible since you know by the above proposition that LOC\*[k] ≤ LOC[k]. Since, we are placing towers at the furthest possible location, the greedy algorithm will stay ahead and make it impossible for m to be greater than n. Thus LOC[] has the same number of towers as the optimal solution and is itself optimal.

* 1. The algorithm complexity is O(n) since I only have to go through the list of houses once and place towers as I traverse the list of the houses.

1. Optimizing Road Placements
2. Place a road 2 miles after the first device. Then keep placing roads 2 miles after a device if the device is not already within 2 miles of a road.

OptimizingRoads (m[1…n])

LOC[1] = m[1] + 2;

k = 2;

For i from 2 to n:

If m[i] is not within the bounds of LOC[k - 1] then

LOC[k] = m[i] + 4

k++

Return LOC;

* 1. Proof of correctness: Placing a road access 2 miles after a device with road access within 2 miles produces the minimum number of access roads.

Setup: Assume LOC\*[k1…kn] is the optimal solution to the access roads placement problem with n number of access roads being placed for coverage of each device and let LOC[k1…km] be the set of access road locations given by our greedy algorithm.

Proposition: Let P(k) be the proposition LOC\*[k] ≤ LOC[k]. Want to show that

this is true for all k1 up to n.

Base Case: Show P(1) is true. LOC\*[1] ≤ LOC[1].

Proof of Base Case: LOC\*[1] ≤ LOC[1] is true since the LOC[1] is placed at the

max range away from m[1], thus LOC\*[1] must be less than

or equal to LOC[1] in order to provide access to the first

device.

Inductive Case: Show that P(k) is true → P(k + 1) is true for any 1≤k<m. Thus

we need to show that LOC\*[k] ≤ LOC[k] → LOC\*[k + 1] ≤ LOC[k + 1].

Proof of Inductive Case:

LOC[k + 1] is the access road location that is 4 miles after the location of

the next device that is not reachable.

Claim: LOC[k + 1] is placed as far down or further down the pipeline than

LOC\*[k + 1].

1. LOC[k + 1] ≥ LOC[k] since we are placing the access roads in increasing order.

2. LOC[k] ≥ LOC\*[k] by the inductive hypothesis

3. LOC[k + 1] ≥ LOC\*[k]

4. LOC\*[k + 1] ≥ LOC\*[k] in order for it to be in the optimal solution,

you cannot have overlapping access roads.

Since, LOC[k + 1] and LOC\*[k + 1] are both ≥ LOC\*[k] and LOC[k + 1] is placed as far down the road possible, in order to reach the device before it, LOC[k + 1] ≥ LOC\*[k + 1]. Thus by Principle of Mathematical Induction, P(k) is true for k = 1..m.

Final Step: Show that m = n. Suppose that m > n. But this is impossible since you know by the above proposition that LOC\*[k] ≤ LOC[k]. Since, we are placing access roads at the furthest possible location, the greedy algorithm will stay ahead and make it impossible for m to be greater than n. Thus LOC[] has the same number of access roads as the optimal solution and is itself optimal.

c. The algorithm complexity is O(n) since I only have to go through the list of devices once and place roads as I traverse the list of the devices.