CS570 Summer 2023
Discussion Week 2

Greedy

Assume the coastline of the country Lyniera is an infinite straight line. There are islands off the coastline of Lyniera. In order to keep a close eye on these islands, the king of Lyniera decided to set up some radar bases along the coastline. Each radar base is a point on the coastline which can cover a circular area around itself with radius d. Let's use the x-axis in a coordinate system as the coastline (horizontal axis), with the sea on the upper side of the x-axis. Each island is a point in the sea with coordinates (x, y). Design a greedy algorithm to find the minimum number of radar bases needed to cover all the islands and give their locations on the coastline. Prove the correctness of your algorithm. (Assume each island is within distance d of the coast, and thus, is possible to be covered by a radar.)

Solution

For each island, draw a circle around it with radius d and we can get an interval on the x-axis where we are able to place a radar base to cover this island (the intersection of the circle with the x-axis). So, translate the input to a set of intervals along the x-axis. The problem is to find the minimum number of points so that each interval has a point inside it.

Greedy strategy:
Put a radar base at the right-most point of the interval with smallest right end-point which has not been covered yet. Remove the intervals which includes the chosen end point. Recursively do the same thing for the remaining intervals until all the intervals are removed.

Proof: Exchange argument

First, prove the first radar base is put correctly. Let the left-most radar base in the optimal solution S* have x-coordinate r*. Then the left-most radar base in our greedy solution S has x-coordinate r $\geq$ r*; otherwise, the optimal solution does not cover this corresponding island. Note that, as we placed our radar base at the right-end point of the interval with smallest right end-point, the set of islands covered by the base at r must include the set of islands covered by base at r*. Thus, we can make an exchange: replace the base at r* in the optimal solution by the base at r, and we form another strategy S', which should also be an optimal solution because still covers all the islands. Therefore, there is an optimal solution that makes the same first choice as the greedy strategy does.

Now consider the small subproblem of covering all islands which have not been covered by our first base. Inductively we can exchange the subsequent bases in S* with the greedy solution while always having an optimal solution that eventually coincides with S when all bases are exchanged. Thus, our greedy algorithm is optimal.

Amortized analysis: Dynamic Array

Start with small array size, say 1. At every insertion:

If array is not full: Insert!
If array is full:

1. New array with double size.
2. Copy existing elements.
3. Insert!

Here is an example:
Init(): capacity=1, length= 0
Add(5): capacity=1, length= 1
Add(7): capacity=2, length= 2
Add(3): capacity=4, length= 3
Add(2), Add(4): capacity=8, length= 5