Luis Enrique Franco Marín

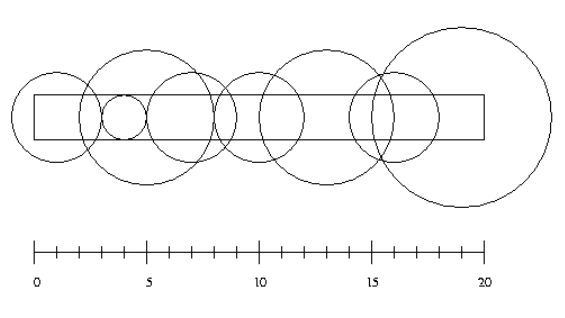
410921353

**Week 8: Interval Cover**

Interval cover represents a greedy algorithm. As explained in the problem it consists of an array of several intervals that may or may not overlap. Given an interval that we want to cover, the algorithm takes the intervals of the array and searches if there is a possible way to overlap a combination of these to cover the test interval.

**Problem:**

Input consists of a number of cases. The first line for each case contains integer number n ,l and w with n≤10000. The next n lines contains two integers giving the position of a sprinkler and its radius of operation. (The picture below illustrates the first case from the sample input.)



For each test case output the minimum number of sprinklers needed to water the entire strip of grass. If it is impossible to water the entire strip output '-1'.

**Code:**

Text

Description automatically generated

**Discussion:**

This is a greedy algorithm. No Dynamic programming nor divide and conquer is required. The problem gives a strip of land and the location of circles with their respective values. The algorithm followed is the one given in the problem description. However, because it’s the position and radius that is given and not the intervals, a previous transformation is necessary. Only the intervals that can actually cover the area are included, that means that any circle with a radius less than half of the width is excluded. The intervals are then modified to be at a distance of sqrt(r^2-(w/2)^2) from the origin. Now that the intervals have been found they are included in an array, this array is at maximum of n size so the space complexity is O(n), the algorithm goes at worst case thru all n intervals but first the intervals need to be sorted, thus the sorting elevates its time complexity to O(n\*lg(n)), then as it compares the interval taken and compares it with others in its worst case is an O(n^2).

Time: O(n^2)

Space: O(n)

**Result:**

A picture containing logo

Description automatically generated