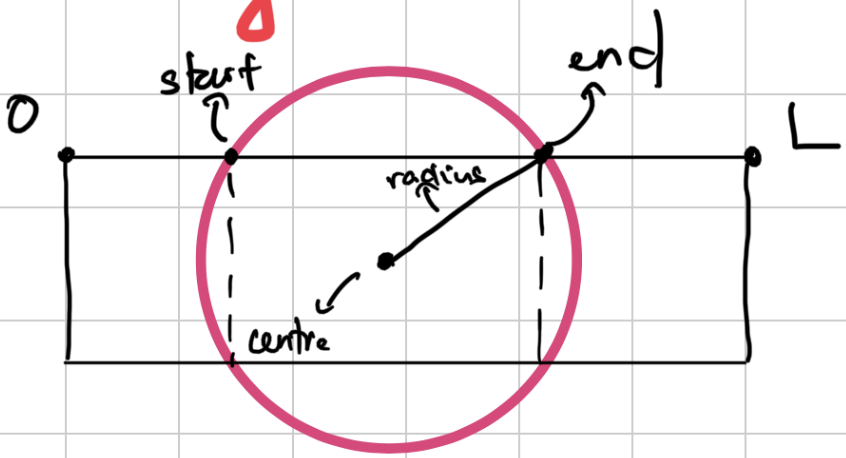
1.3 Algorithm



Recursion Idea:

- Sort Sprikler by start point (natural order), then by radius (reverse order).

so that we always "stay ahead"

- Given n sprinklers:

MaxArea(c, k) = the max area (starting from 0.0) that can be covered by k sprinkers in the first c sprinklers ($k \le c$)

e.g: MaxArea(2, 4) = 3.0

↳ The max area that can be covered by any 2 sprinklers in the first 4 sprinklers is 3.0

- Recursion:

k sprinklers in the first c-1 sprinkler ⇒ not include c^{th} sprinkler

$$MaxArea(c, k) = Max\left(\underbrace{MaxArea(c-1, k)}, \; MaxArea(c, k-1) + \text{Sprinkler } c^{th}\right)$$

... in the first 4 sprinklers is 3.0

- **Recursion:**

k sprinklers in the first c−1 sprinkler ⇒ not include c^{th} sprinkler

$$\text{MaxArea}(c, k) = \text{Max}\begin{pmatrix}\text{MaxArea}(c-1, k), \\ \text{MaxArea}(c, k-1) + \text{Sprinkler } c^{th}\end{pmatrix}$$

k−1 sprinklers in the first c sprinklers (may or may not include c^{th} sprinkler)

"exchange" only include c^{th} sprinkler if it gives better result

- **Use memoization technique**

L = 5, 0

c	Sprinklers \ k	1	2	3
1	[0, 3]	3	3	3
2	[0, 2]	3	3	3
3	[2, 4]	3	4	4
4	[4, 5]	3	4	5

sorted

start end

$$\text{table}[3,2] = \text{Max}\begin{pmatrix}\text{table}[2,2], \\ \text{table}[3,1] + 3^{rd} \text{ sprinkler}\end{pmatrix}$$

$$= \text{Max}\begin{pmatrix}\text{area from } 0 \to 3, \\ \underbrace{\text{area from } 0 \to 3 + \text{area from } 2 \to 4}_{\text{contiguous} \to \text{area from } 0 \to 4}\end{pmatrix}$$

	[0, 3]	3	3	3
2	[0, 2]	3	3	3
3	[2, 4]	3	4	4
4	[4, 5]	3	4	5

sorted

start end

table [3,2] = Max (table [2,2],
table [3, 1] + 3rd sprinkler)

= Max (area from 0 → 3 ,
area from 0 → 3 + area from 2 → 4)

$\underbrace{\text{area from } 0 \to 3 + \text{area from } 2 \to 4}_{\text{contiguous} \to \text{area from } 0 \to 4}$

= area from 0 → 4
= 4

table [4,2] = Max (table (3,2),
table (4,1) + 4th sprinkler)

= Max (area from 0 → 4 ,
area from 0 → 3 + area from 4 → 5)

$\underbrace{\text{area from } 0 \to 3 + \text{area from } 4 \to 5}_{\text{not contiguous} \Rightarrow \text{area from } 0 \to 3}$

= area from 0 → 4

ALGORITHM

```
* findMinimum (Sprinklers, L, W):
    SortSprinklers ()
    Verify AllSprinklers CanCover the Garden ()
    table = array [n. of sprinklers][n. of sprinklers]
    # Populate the 1st column & 1st row with
      the frst sprikler's end.
    for i in range(1, n. of sprinkler):
        table [0, i] = 1st sprinker's end.
        table [i, 0] = 1st sprinker's end
    if sprinkler's end ≥ L:
        return 1
    for k in range (2, n. of sprinkler):
        for c in range (2, n. of sprinkler):
            a = table [c-1, k]
            b = table [c, k-1] + sprinklers[c]
            table [c, k] = max (a, b)
            if table [c, k] ≥ L:
                return k
    return 0;
```