Agenda:

- Contiguous memory allocation in Arrays
- Bucketing Technique:
    - Smallest Positive missing number
    - Boolean Matrix Question
- Kadane's Algorithm
- Max Circular Subarray Sum (Just discuss the approach)
- Prefix Sum Technique:
    - Maximum Occurred Integer in N Ranges

Smallest Positive missing number

You are given an array **arr**[] of **N** integers including 0. The task is to find the smallest positive number missing from the array.

Example 1:

```
Input:
N = 5
arr[] = {1,2,3,4,5}
Output: 6
Explanation: Smallest positive missing
number is 6.
```

Example 2:

```
Input:
N = 5
arr[] = {0,-10,1,3,-20}
Output: 2
Explanation: Smallest positive missing
number is 2.
```

A Boolean Matrix Question

Given a matrix, A of size M x N of 0s and 1s. If an element is 0, set its entire row and column to 0.

```
Input 1:
    [   [1, 0, 1],
        [1, 1, 1],
        [1, 1, 1]   ]


Output 1:
    [   [0, 0, 0],
        [1, 0, 1],
        [1, 0, 1]   ]


Input 2:
    [   [1, 0, 1],
        [1, 1, 1],
        [1, 0, 1]   ]


Output 2:
    [   [0, 0, 0],
        [1, 0, 1],
        [0, 0, 0]   ]
```

Kadane's Algorithm

Given an array **Arr[]** of **N** integers. Find the contiguous sub-array(containing at least one number) which has the maximum sum and return its sum.

```
Input:
N = 5
Arr[] = {1,2,3,-2,5}
Output:
9
Explanation:
Max subarray sum is 9
of elements (1, 2, 3, -2, 5) which
is a contiguous subarray.
```

Example 2:

```
Input:
N = 4
Arr[] = {-1,-2,-3,-4}
Output:
-1
Explanation:
Max subarray sum is -1
of element (-1)
```

Maximum Occurred Integer In N Ranges

Given **n** integer ranges, the task is to find the maximum occurring integer in these ranges. If more than one such integer exits, find the smallest one. The ranges are given as two arrays **L**[] and **R**[]. L[i] consists of starting point of range and R[i] consists of corresponding end point of the range.

For example consider the following ranges.
L[] = {2, 1, 3}, R[] = {5, 3, 9)
Ranges represented by above arrays are.
[2, 5] = {2, **3**, 4, 5}
[1, 3] = {1, 2, **3**}
[3, 9] = {**3**, 4, 5, 6, 7, 8, 9}
The maximum occurred integer in these ranges is 3.

Input:
n = 4
L[] = {1,4,3,1}
R[] = {15,8,5,4}
Output: 4
Explanation: The given ranges are [1,15]
[4, 8] [3, 5] [1, 4]. The number that
is most common or appears most times in
the ranges is 4.

Constraints:
$1 \le n \le 10^6$
$0 \le L[i], R[i] \le 10^6$

(IMPORTANT)