NUMBER THAT APPEARS ONCE

☆ In this problem, each number within the given array will appear twice except once. That number is to be returned.

Brute force solution is to do a linear search for each element and breaking out once an element only shows up once.
Better solution involves the use of a hasharray of size (max element +1) or a hashmap. This changes time complexity from $O(N^2)$ to $O(3N)$ but adds space complexity.
Hasharray doesn't work for negative elements or very large elements. so map is better. Another thing we could do is to create array of pairs, size will be half the size of main array.
Optimal solution is to XOR all the elements and the element which appears once will be left.

Pseudocode :

```
numberThatAppearsOnce(arr, N) {
    int x = 0
    for(int i = 0 → N) {
        x = x ^ arr[i]
    }
```

```
    return x
}
```