Single Number

Given a non-empty array of integers nums, every element appears twice except for one. Find that single element.

Yaa

Ex:

Input: nums = [2, 2, 1]

output: 1

Input: nums = [4, 1, 2, 1, 2]

output: 4

constraints:

- 1 <= nums.length <= 3 * 10^4
- -3 * 10^4 <= num[i] <= 3 * 10^4
- Each element in array appear twice except for one which appears once

Algorithm

1) Initialize result = 0
2) Iterate through each element in array and perform XOR with result.
   - Numbers that apper twice cancel out (x ^ x = 0)
   - Numbers that appear once remains in result
3) Return result

How it works:

Nums = [4, 1, 2, 1, 2]

result = 4 ^ 1 ^ 2 ^ 1 ^ 2

= 4 ^ (1 ^ 1) ^ (2 ^ 2)

= 4 ^ 0 ^ 0

= 4.

Dry run:

nums = [4, 1, 1, 4]

i = 0

=> result = 0

=> result = result ^ nums[i]

result = 0 ^ 4

result = 4

i++ => i = 1

result = 4

=> result = 4 ^ 1

result = 5

i++ => i = 2

result = 5 ^ 2

result = 4 => return 4

ode:

```
class Solution {
    public int singleNumber(int[] nums) {
        int result = 0; // stores element to return
        for (int i = 0; i < nums.length; i++) {
            // iterates through array elements
            result ^= nums[i]; // cancels out pairs
        }
        return result;
    }
}
```

Time complexity: O(n)

Space complexity: O(1)