Single Number II

Question: Given an array of integers, every element appears three times except for one. Find that single one.

Note: Your algorithm should have a linear runtime complexity. Could you implement it without using extra memory?

Solutions:

```
class Solution:
    # @param A, a list of integer
    # @return an integer
    def singleNumber(self, A):
        bit = [0 for i in range(32)]
        for number in A:
            for i in range(32):
                if (1 << i) & number == 1 << i: bit[i] += 1
        res = 0
        if bit[31] % 3 == 0:
            for i in range(31):
                if bit[i] % 3 == 1: res += 1 << i
        else:
            for i in range(31):
                if bit[i] % 3 == 0: res += 1 << i
            res = -(res + 1)
        return res


Solution().singleNumber([1, 2, 1, 2, 1, 2, 0, 0])
```