# **≡** Articles > 136. Single Number ▼

Aug. 31, 2017 | 534.6K views

136. Single Number

\*\*\*\* Average Rating: 4.85 (255 votes)

**f y in** 

🖺 Сору

🖺 Сору

🖺 Сору

🖺 Сору

Post

Note:

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

### Your algorithm should have a linear runtime complexity. Could you implement it without using extra

memory? Example 1:

### Input: [2,2,1]

Output: 1

```
Example 2:
 Input: [4,1,2,1,2]
```

```
Output: 4
```

## Approach 1: List operation

Algorithm

Solution

### 1. Iterate over all the elements in nums 2. If some number in nums is new to array, append it

## 3. If some number is already in the array, remove it

4

Python Java

:type nums: List[int]

:rtype: int

class Solution(object): def singleNumber(self, nums):

```
no_duplicate_list = []
             for i in nums:
                 if i not in no_duplicate_list:
                     no_duplicate_list.append(i)
  10
                 else:
  11
                     no_duplicate_list.remove(i)
  12
             return no_duplicate_list.pop()
  13
Complexity Analysis
   ullet Time complexity : O(n^2). We iterate through nums, taking O(n) time. We search the whole list to
     find whether there is duplicate number, taking O(n) time. Because search is in the for loop, so we
     have to multiply both time complexities which is O(n^2).
```

Approach 2: Hash Table

## 2. Return the element which appeared only once.

Java

Python

**Algorithm** 

from collections import defaultdict class Solution:

We use hash table to avoid the O(n) time required for searching the elements.

1. Iterate through all elements in **nums** and set up key/value pair.

• Space complexity : O(n). We need a list of size n to contain elements in nums.

for i in nums: hash\_table[i] += 1

def singleNumber(self, nums: List[int]) -> int:

for i in hash\_table: if hash\_table[i] == 1:

hash\_table = defaultdict(int)

```
return i
 10
Complexity Analysis
   • Time complexity : O(n \cdot 1) = O(n). Time complexity of for loop is O(n). Time complexity of hash
     table(dictionary in python) operation pop is O(1).
   • Space complexity : O(n). The space required by hash\_table is equal to the number of elements in
     nums.
```

## Approach 3: Math Concept

Python

**Complexity Analysis** 

class Solution(object):

:rtype: int

Java

2\*(a+b+c) - (a+a+b+b+c) = c

def singleNumber(self, nums):

:type nums: List[int]

return 2 \* sum(set(nums)) - sum(nums)

of elements(n) in nums. • Space complexity : O(n+n)=O(n). set needs space for the elements in nums Approach 4: Bit Manipulation Concept • If we take XOR of zero and some bit, it will return that bit  $\circ a \oplus 0 = a$ • If we take XOR of two same bits, it will return 0

• Time complexity : O(n+n)=O(n). sum will call next to iterate through nums. We can see it as

sum(list(i, for i in nums)) which means the time complexity is O(n) because of the number

:type nums: List[int] :rtype: int

> for i in nums: a ^= i

def singleNumber(self, nums):

•  $a \oplus b \oplus a = (a \oplus a) \oplus b = 0 \oplus b = b$ 

So we can XOR all bits together to find the unique number.

 $\circ \ a \oplus a = 0$ 

class Solution(object):

a = 0

return a

elements in nums.

Preview

Python

**Complexity Analysis** 

Java

10

```
• Space complexity : O(1).
Rate this article: * * * * *
 O Previous
                                                                                            Next 1
Comments: 178
                                                                                           Sort By ▼
            Type comment here... (Markdown is supported)
```

• Time complexity : O(n). We only iterate through nums, so the time complexity is the number of

```
supercow ★ 948 ② January 11, 2019 2:35 PM
how to train my brain to come with solution 4?
941 A V Share  Reply
SHOW 29 REPLIES
yingge ★ 276 ② January 16, 2019 5:12 PM
Seeing the solution 4, maybe is the difference between the intelligent and me.
273 \Lambda 🗸 🗗 Share 🦘 Reply
SHOW 7 REPLIES
richardli515 ★ 165 ② September 23, 2018 11:10 AM
Who contributed to this solution article? It's so concise in concept/algorithm descriptions and the
article length doesn't affect its easiness to understand. It makes it even clearer with the bullet points.
And the XOR algorithm just used 3 bullet points to successfully explain an important and not so
straightforward trick in coding interviews. I think the author should publish a book to explain all the
other coding problems in this way.
                                            Read More
148 A V 🗗 Share 🦘 Reply
SHOW 2 REPLIES
nojuma70 ★ 108 ② July 3, 2018 6:13 AM
first 3 approaches i figured them out. made me feel good. approach 4 made me realize i know nothing.
108 ∧ ∨ ♂ Share ★ Reply
SHOW 3 REPLIES
PerTayse ★ 88 ② April 2, 2019 9:57 AM
I just wanna kiss the one who came up with solution 4!!!!
88 \Lambda 🗸 🗗 Share 🦙 Reply
SHOW 4 REPLIES
EddieDong ★ 151 ② July 5, 2018 7:01 AM
only approach 4 fulfills the requirement of the problem which require O(n) time complexity and O(1)
space.
48 \Lambda 🗸 🖸 Share 🦙 Reply
lostandfoundii ★ 182 ② June 16, 2018 5:12 AM
OMG Approach 4!!!!!
45 A V C Share    Reply
fuckyoudeletemyaccount ★ 37 ② September 1, 2018 1:48 PM
Approach 4 can be further 'reduced' to 1 line:
 return reduce(lambda x, y: x ^ y, nums)
29 A V C Share  Reply
SHOW 7 REPLIES
foxyz ★ 330 ② February 25, 2019 1:40 AM
XOR solution is sol cool!
23 🔨 🔀 Share 👆 Reply
SHOW 1 REPLY
terrible_whiteboard ★ 627 ② May 19, 2020 6:09 PM
I made a video if anyone is having trouble understanding the solution (clickable link)
```

Read More

https://youtu.be/m8QvwmfJwvA

21 A V C Share Reply

( 1 2 3 4 5 6 ... 17 18 >