

**NOME:** Caroline Souza Camargo

**REGRA DE CÓDIGO LEGÍVEL:** Stanford Style Guide

**EXERCÍCIO:** Trabalho Prático 4

Dado um vetor `nums` contendo  $n$  números inteiros positivos e únicos encontre o número faltante na sequência. Assuma que somente um número está faltando para completar a sequência numérica.

A complexidade da resposta não pode ser superior a  $O(n \log n)$  Dica, existe uma solução  $O(n)$

Exemplo 1:

Entrada: `nums = [3, 0, 1]`

Saída: 2

Explicação:  $n = 3$  já que existem 3 números, então todos os números estão no intervalo  $[0,3]$ . 2 é o número ausente no intervalo, pois não aparece em `nums`.

Exemplo 2:

Entrada: `nums = [0, 1]`

Saída: 2

Explicação:  $n = 2$  já que existem 2 números, então todos os números estão no intervalo  $[0,2]$ . 2 é o número ausente no intervalo, pois não aparece em `nums`.

Exemplo 3:

Entrada: `nums = [9,6,4,2,3,5,7,0,1]`

Saída: 8

Explicação:  $n = 9$  já que existem 9 números, então todos os números estão no intervalo  $[0,9]$ . 8 é o número ausente no intervalo, pois não aparece em `nums`.

**LINK PARA O GIT:**

<https://github.com/Caroline-Camargo/AlgoritimoseEstruturaDeDados/tree/master/TrabalhoPratico4>

**LINK PARA O VÍDEO:**

[https://drive.google.com/file/d/17a2eNAqN501LDReiGjzp7f4jDS\\_6ZFkC/view?usp=sharing](https://drive.google.com/file/d/17a2eNAqN501LDReiGjzp7f4jDS_6ZFkC/view?usp=sharing)

**O QUE FUNCIONA:** Após vários testes com várias entradas e testando com o debugador e o Dr Memory o programa funcionou corretamente

## O QUE NÃO FUNCIONA:

Durante o período da aula foi esquecido de alguns detalhes e alguns erros foram encontrados no código

- Função MergeSort tinha dois erros de inicialização de variáveis com valores errados
  - O código possuía um bug em dizer qual era o número faltante na sequência quando a pessoa digitava um número apenas
  - Não tinha dado tempo de desenvolver uma função para verificar se o vetor digitado possuía elementos repetidos, uma vez que era dito que o vetor nums deveria possuir números inteiros positivos e únicos
- Todos problemas foram resolvidos

## LISTA DE TESTES:

**Teste 1:** [3, 0, 1]  
Saída: 2

**Teste 2:** [0, 1]  
Saída: 2

**Teste 3:** [9,6,4,2,3,5,7,0,1]  
Saída: 8

**Teste 4:** [35]  
Saída: 36

**Teste 5:** [102, 99, 100]  
Saída: 101

**Teste 6:** [25, 30, 26, 27, 33, 31, 28, 32, 24]  
Saída: 29

**Teste 7:** [19,8,11,9,17,6,13,2,14,3,16,5,15,7,12,1,10,18]  
Saída: 4

**Teste 8:** [997,998,1000,998]  
Saída: Erro a entrada deve ser única

## Print do Dr Memory com o Exercício desenvolvido:

```
results.txt - Bloco de notas
Arquivo  Editar  Exibir  ⚙️

SUPPRESSIONS USED:

ERRORS FOUND:
  0 unique,  0 total unaddressable access(es)
  3 unique, 11 total uninitialized access(es)
  0 unique,  0 total invalid heap argument(s)
  0 unique,  0 total GDI usage error(s)
  0 unique,  0 total handle leak(s)
  0 unique,  0 total warning(s)
  0 unique,  0 total,  0 byte(s) of leak(s)
  0 unique,  0 total,  0 byte(s) of possible leak(s)
ERRORS IGNORED:
  7 unique,  7 total, 4566 byte(s) of still-reachable allocation(s)
  (re-run with "-show_reachable" for details)
Details: C:\Users\carol\AppData\Roaming\Dr. Memory\DrMemory-
main.exe.8028.000\results.txt

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8
```

## Print do Dr Memory com um Olá mundo:

```
results.txt - Bloco de notas
Arquivo  Editar  Exibir  ⚙️

SUPPRESSIONS USED:

ERRORS FOUND:
  0 unique,  0 total unaddressable access(es)
  3 unique, 11 total uninitialized access(es)
  0 unique,  0 total invalid heap argument(s)
  0 unique,  0 total GDI usage error(s)
  0 unique,  0 total handle leak(s)
  0 unique,  0 total warning(s)
  0 unique,  0 total,  0 byte(s) of leak(s)
  0 unique,  0 total,  0 byte(s) of possible leak(s)
ERRORS IGNORED:
  7 unique,  7 total, 4566 byte(s) of still-reachable allocation(s)
  (re-run with "-show_reachable" for details)
Details: C:\Users\carol\AppData\Roaming\Dr. Memory\DrMemory-
main.exe.3124.000\results.txt

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8
```