

## 查找一个连续数组中缺失的数字

连续整数序列  $[1, 2, \dots, n]$  的和可以用公式  $(n * (n + 1)) / 2$  来计算, 计为 `expectedSum`。

数组中所有元素的和可以用 `arr.reduce((sum, num) => sum + num, 0)` 来计算 `actualSum`。

`expectedSum - actualSum` 便得到缺失的数字。

```
function findMissingNum(arr) {  
  // 取出数组中最大的数  
  let maxNum = Math.max(...arr);  
  
  // 计算连续数组的元素和 (不缺失元素情况下)  
  let expectedSum = maxNum * (maxNum + 1) / 2;  
  
  // 计算数组实际所有元素的和  
  let actualSum = arr.reduce((sum, num) => sum + num, 0);  
  
  // 寻找丢失的数字  
  let missingNum = expectedSum - actualSum;  
  
  return missingNum;  
}
```

```
// 测试  
let arr1 = [3, 4, 5, 9, 1, 2, 8, 7]; // 缺少 6 (没有按照递增、递减顺序排列)  
  
let arr2 = [9, 3, 2, 6, 8, 7, 1, 5]; // 缺少 4 (没有按照递增、递减顺序排列)  
  
findMissingNum(arr1); // 6  
findMissingNum(arr2); // 4
```