链滴

什么是大 O 符号?

作者：Vanessa

原文链接：https://ld246.com/article/1549713468969

来源网站：链滴

2019-02-09

回答

大 O 符号在计算机科学中 用来描述算法的时间复杂度。执行速度快且复杂性低的算法视为优秀的算
。
算法的运行次数并不是每次都相同，大部分取决于所提供的数据。在某些情况下，他们执行的很快，某些情况下，他们却执行的很慢（哪怕他们的数据是一样多）。

以下示例中，我们假设基准时间为：1element = 1ms

O(1)

```
arr[arr.length - 1] // 1000 elements = 1ms
```

时间复杂度恒定。无论数组有多少元素，理论（不考虑机器性能、当前环境等因素）上他执行的时间量是相同的。

O(N)

```
arr.filter(fn) // 1000 elements = 1000ms
```

线性时间复杂度。执行时间将随数组元素个数呈线性增加。如果数组拥有 1000 个元素且函数运行需花费 1ms，那么 7000 个元素需要执行 7ms。这是因为函数在返回结果之前必须迭代数组中的所有素。

O([1, N])

```
arr.some(fn) // 1000 elements = 1ms <= x <= 1000ms
```

执行时间的长短取决于提供给函数的数据，他需要的时间可能很短，也可能很长。最好的情况是 O(1) 最坏的情况是 O(N)。

O(NlogN)

```
arr.sort(fn) // 1000 elements ~= 10000ms
```

浏览器通常为 sort() 方法使用快速排序算法进行实现，快速排序的平均时间复杂度为O(NlogN)。这于数据很多的集合非常有效。

O(N^2)

```
for (let i = 0; i < arr.length; i++) {
  for (let j = 0; j < arr.length; j++) {
    // 1000 elements = 1000000ms
  }
}
```

执行时间随元素数量呈二次方增长。这通常是由于使用了嵌套循环。

O(N!)

```
// 1000 elements = Infinity ms
const permutations = arr => {
  if (arr.length <= 2) return arr.length === 2 ? [arr, [arr[1], arr[0]]] : arr
  return arr.reduce(
    (acc, item, i) =>
      acc.concat(
        permutations([...arr.slice(0, i), ...arr.slice(i + 1)]).map(val => [
          item,
          ...val
        ])
      ),
    []
  )
}
```

数组中即使只增加一个元素，也会使执行时间增加的非常长。

加分回答

- 嵌套循环的执行时间会随着元素的增长呈指数增长，因此遇到嵌套循环需考虑到性能问题。

返回总目录

每天 30 秒