

198. 打家劫舍

难度 中等  1857  收藏  分享  切换为英文  接收动态  反馈

你是一个专业的小偷，计划偷窃沿街的房屋。每间房内都藏有一定的现金，影响你偷窃的唯一制约因素就是相邻的房屋装有相互连通的防盗系统，如果两间相邻的房屋在同一晚上被小偷闯入，系统会自动报警。

给定一个代表每个房屋存放金额的非负整数数组，计算你 **不触动警报装置**的情况下，一夜之内能够偷窃到的最高金额。

示例 1:

输入: [1,2,3,1]
输出: 4
解释: 偷窃 1 号房屋 (金额 = 1)，然后偷窃 3 号房屋 (金额 = 3)。
偷窃到的最高金额 = 1 + 3 = 4。

示例 2:

输入: [2,7,9,3,1]
输出: 12
解释: 偷窃 1 号房屋 (金额 = 2)，偷窃 3 号房屋 (金额 = 9)，接着偷窃 5 号房屋 (金额 = 1)。
偷窃到的最高金额 = 2 + 9 + 1 = 12。

关键， $f(k)$ 也可以由 $f(k-1)$ 和 $f(k-2)$ 求出

应该是拆分成两个情况看看有没有被排除在外的，没有就可以这么做

213. 打家劫舍 II

难度 中等 908 收藏 分享 切换为英文 接收动态 反馈

你是一个专业的小偷，计划偷窃沿街的房屋。每间房内都藏有一定的现金。这个地方所有的房屋都围成一圈，这意味着第一个房屋和最后一个房屋是紧挨着的。同时，相邻的房屋装有相互连通的防盗系统，如果两间相邻的房屋在同一晚上被小偷闯入，系统会自动报警。

给定一个代表每个房屋存放金额的非负整数数组，计算你 在不触动警报装置的情况下，今晚能够偷窃到的最高金额。

示例 1:

```
输入: nums = [2,3,2]
输出: 3
解释: 你不能先偷窃 1 号房屋（金额 = 2），然后偷窃 3 号房屋（金额 = 2），因为他们是相邻的。
```

示例 2:

```
输入: nums = [1,2,3,1]
输出: 4
解释: 你可以先偷窃 1 号房屋（金额 = 1），然后偷窃 3 号房屋（金额 = 3）。
偷窃到的最高金额 = 1 + 3 = 4。
```

示例 3:

```
输入: nums = [1,2,3]
输出: 3
```

总体思路:

- 此题是 198. 打家劫舍 的拓展版：唯一的区别是此题中的房间是环状排列的（即首尾相接），而 198. 题中的房间是单排排列的；而这也是此题的难点。
- 环状排列意味着第一个房子和最后一个房子中只能选择一个偷窃，因此可以把此环状排列房间问题约化为两个单排排列房间子问题：
 - 在不偷窃第一个房子的情况下（即 $nums[1:]$ ），最大金额是 p_1 ；
 - 在不偷窃最后一个房子的情况下（即 $nums[:n-1]$ ），最大金额是 p_2 。

综合偷窃最大金额：为以上两种情况的较大值，即 $\max(p_1, p_2)$ 。
- 下面的任务则是解决 单排排列房间（即 198. 打家劫舍）问题。推荐可以先把 198. 做完再做这道题。

740. 删除并获得点数

难度 中等  534  收藏  分享  切换为英文  接收动态  反馈

给你一个整数数组 `nums`，你可以对它进行一些操作。

每次操作中，选择任意一个 `nums[i]`，删除它并获得 `nums[i]` 的点数。之后，你必须删除所有等于 `nums[i] - 1` 和 `nums[i] + 1` 的元素。

开始你拥有 0 个点数。返回你能通过这些操作获得的最大点数。

示例 1:

输入: `nums = [3,4,2]`

输出: 6

解释:

删除 4 获得 4 个点数，因此 3 也被删除。

之后，删除 2 获得 2 个点数。总共获得 6 个点数。

示例 2:

输入: `nums = [2,2,3,3,3,4]`

输出: 9

解释:

删除 3 获得 3 个点数，接着要删除两个 2 和 4。

之后，再次删除 3 获得 3 个点数，再次删除 3 获得 3 个点数。

总共获得 9 个点数。

举个例子:

```
nums = [2, 2, 3, 3, 3, 4]
```

构造后:

```
all=[0, 0, 2, 3, 1];
```

就是代表着 2 的个数有两个，3 的个数有 3 个，4 的个数有 1 个。

其实这样就可以变成打家劫舍的问题了呗。

我们来看看，打家劫舍的最优子结构的公式:

```
dp[i] = Math.max(dp[i - 1], dp[i - 2] + nums[i]);
```

再来看看现在对这个问题的最优子结构公式:

```
dp[i] = Math.max(dp[i - 1], dp[i - 2] + i * all[i]);
```