

A4_README

第一题

算法思路

定义 count 计算 flowerbed 中最多可以种植的位置的数量，与 n 比较，若 count \geq n 则返回 true，否则返回 false。

count 的计算方法：遍历 flowerbed 数组，若当前元素值为 0，且左右两个元素值均为 0，则 count 加 1，并将该元素值置为 1；若当前元素是数组的头或尾元素，则仅需考虑其右或左一个元素即可。

测试用例

```
Please enter the size of flowerbed: 5
Please enter flowerbed: 1 0 0 0 1
Please enter n: 1
true
```

```
Please enter the size of flowerbed: 5
Please enter flowerbed: 1 0 0 0 1
Please enter n: 2
false
```

```
Please enter the size of flowerbed: 5
Please enter flowerbed: 1 0 1 0 1
Please enter n: 1
false
```

第二题

算法思路

采用动态规划方法。用 dp1[i] 和 dp2[j] 来分别记录遍历到 nums1[i] 和 nums2[j] 时的最大得分。

初始化：dp1 与 dp2 最初都为 0。然后，从左到右遍历 nums1 和 nums2。

若 nums1[i] 小于 nums2[j]，则需在 nums1 中前进，并更新 dp1，即 $dp1[i] = dp1[i - 1] + nums1[i]$ 。

若 nums1[i] 大于 nums2[j]，则需在 nums2 中前进，并更新 dp2，即 $dp2[j] = dp2[j - 1] + nums2[j]$ 。

若 nums1[i] 等于 nums2[j]，则可在 nums1 或 nums2 中前进，同时更新 dp1 和

dp2, 即 $dp1[i] = dp2[j] = \max(dp1[i - 1], dp2[j - 1]) + \text{nums1}[i]$ 。

遍历完成后, dp1 和 dp2 中较大的那个就是最终结果。

此算法的时间复杂度为 $O(n1 + n2)$, 其中 $n1$ 和 $n2$ 是两个数组的长度, 因为每个数组的每个元素最多被访问一次。

测试用例

```
Please enter the size of nums1: 5
Please enter nums1: 2 4 5 8 10
Please enter the size of nums2: 4
Please enter nums2: 4 6 8 9
30
```

```
Please enter the size of nums1: 5
Please enter nums1: 1 3 5 7 9
Please enter the size of nums2: 3
Please enter nums2: 3 5 100
109
```

```
Please enter the size of nums1: 5
Please enter nums1: 1 2 3 4 5
Please enter the size of nums2: 5
Please enter nums2: 6 7 8 9 10
40
```