

假设你有一个很长的花坛，一部分地块种植了花，另一部分却没有。可是，花卉不能种植在相邻的地块上，它们会争夺水源，两者都会死去。

给定一个花坛（表示为一个数组包含0和1，其中0表示没种植花，1表示种植了花），和一个数 **n**。能否在不打破种植规则的情况下种入 **n** 朵花？能则返回True，不能则返回False。

示例 1:

输入: flowerbed = [1,0,0,0,1], n = 1

输出: True

示例 2:

输入: flowerbed = [1,0,0,0,1], n = 2

输出: False

注意:

1. 数组内已种好的花不会违反种植规则。
2. 输入的数组长度范围为 [1, 20000]。
3. **n 是非负整数，且不会超过输入数组的大小。**

分析，典型贪婪算法题目，若是栽就栽，并种花个数+1，然后动态统计最大的种花个数，若是n小于等于最大个数，返回true，否则，返回false

```
class Solution {
public:
    bool canPlaceFlowers(vector<int>& flowerbed, int n) {
        if(flowerbed.size()==1){
            int num=0;
            if(flowerbed[0]==0){
                num++;
            }
            if(n<=num){
                return true;
            }
            return false;
        }
        int num=0;
        for(int i=0;i<flowerbed.size();i++){
            if(i==0){
```

```
        if (flowerbed[i]==0&&flowerbed[i+1]==0) {
            num++;
            flowerbed[i]=1;
        }
    }else if(i==flowerbed.size()-1){
        if (flowerbed[i]==0&&flowerbed[i-1]==0) {
            num++;
            flowerbed[i]=1;
        }
    }else{
        if (flowerbed[i-1]==0&&flowerbed[i]==0&&flowerbed[i+1]==0) {
            num++;
            flowerbed[i]=1;
        }
    }
}

if (n<=num) {
    return true;
}

return false;
}

};
```