


605. Can Place Flowers

Solved 

Easy

Topics

Companies

You have a long flowerbed in which some of the plots are planted, and some are not. However, flowers cannot be planted in **adjacent** plots.

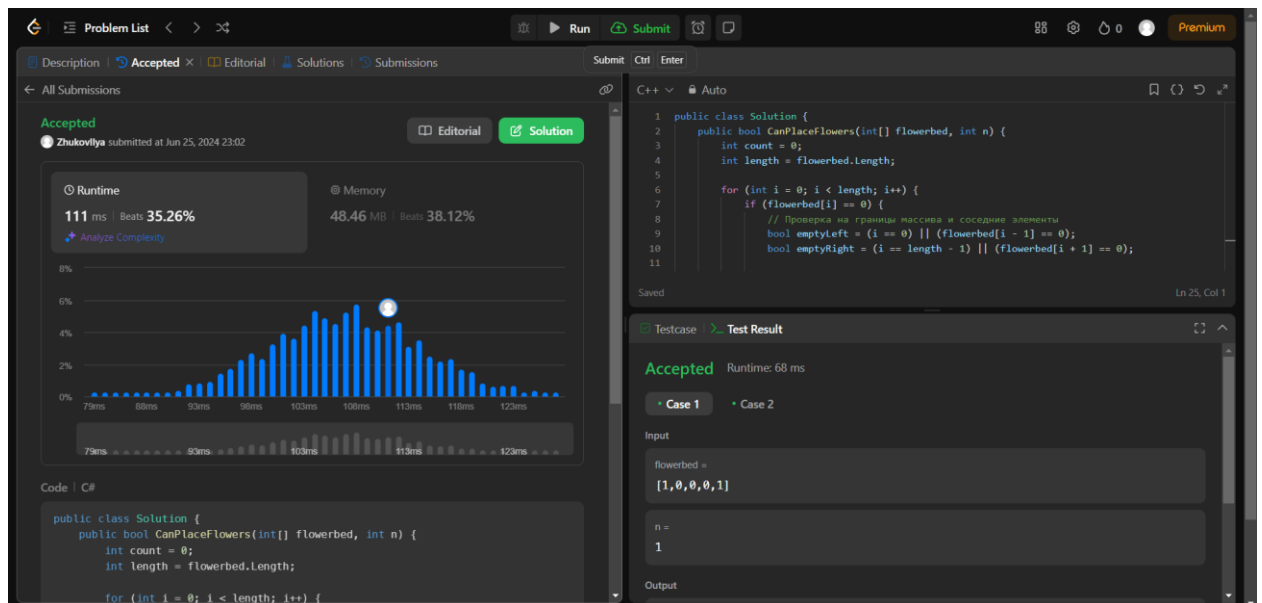
Given an integer array `flowerbed` containing `0`'s and `1`'s, where `0` means empty and `1` means not empty, and an integer `n`, return `true` if `n` new flowers can be planted in the `flowerbed` without violating the no-adjacent-flowers rule and `false` otherwise.

Example 1:

Input: `flowerbed = [1,0,0,0,1]`, `n = 1`
Output: `true`

Example 2:

Input: `flowerbed = [1,0,0,0,1]`, `n = 2`
Output: `false`



The screenshot displays a code editor interface for the 'Can Place Flowers' problem. The code is written in C++ and defines a class `Solution` with a method `CanPlaceFlowers`. The method iterates through the `flowerbed` array and checks for adjacent flowers. The test result shows 'Accepted' with a runtime of 68 ms.

```
public class Solution {
    public bool CanPlaceFlowers(int[] flowerbed, int n) {
        int count = 0;
        int length = flowerbed.Length;

        for (int i = 0; i < length; i++) {
            if (flowerbed[i] == 0) {
                // Проверка на границы массива и соседние элементы
                bool emptyLeft = (i == 0) || (flowerbed[i - 1] == 0);
                bool emptyRight = (i == length - 1) || (flowerbed[i + 1] == 0);
                if (emptyLeft & emptyRight) {
                    count++;
                }
            }
        }

        return count >= n;
    }
}
```

Код:

```
public class Solution
{
    public bool CanPlaceFlowers(int[] flowerbed, int n)
    {
        int count = 0;
        int length = flowerbed.Length;

        for (int i = 0; i < length; i++)
        {
            if (flowerbed[i] == 0)
            {
                // Проверка на границы массива и соседние элементы
                bool emptyLeft = (i == 0) || (flowerbed[i - 1] == 0);
```

```
bool emptyRight = (i == length - 1) || (flowerbed[i + 1] == 0);

if (emptyLeft && emptyRight)
{
    flowerbed[i] = 1;
    count++;
    if (count >= n)
    {
        return true;
    }
}

return count >= n;
}
```