

Can Place Flowers

Question:

<https://leetcode.com/problems/can-place-flowers/>

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 if `n` new flowers can be planted in the `flowerbed` without violating the no-adjacent-flowers rule.

Example 1:

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

My approach:

My approach was to see if any index `i` has a 0 to its left and to its right. If so then we place a flower at index `i`.

But the problem was that the boundary cases couldn't be solved using this approach and needed separate handling.

Boundary Cases:

1. Leading Zeros([0,0,1,...] we can place a flower at index=0)
2. `n==0`
3. Only 1 value in the array.
4. Trailing zeros,i.e no 1 at the end([...,1,0,0] we can place a flower at index=`n-1`)

My Solution:

<https://gist.github.com/vermaayush680/9969f23956c105acbad3bb6d75dc732a>

if `n==0`:

```

    return True
if len(f)==1:
    if f[0]==0 and n==1:
        return True
    else:
        return False

if f[0]==0 and f[1]==0:
    f[0]=1
    n-=1
if n==0:
    return True

i=1
while(n>0 and i<(len(f)-1)):
    if f[i]==0 and f[i-1]==0 and f[i+1]==0:
        f[i]=1
        n-=1
    i+=1

if f[len(f)-1]==0 and f[len(f)-2]==0 and n>0:
    f[n-1]=1
    n-=1

if n==0:
    return True
else:
    return False

```

Condensed Approach:

We add a 0 at the beginning and the end and this solves all the boundary cases directly without special handling.

[0,0,1,....] becomes [0,0,0,1,....]

[.....,1,0,0] becomes [.....,1,0,0,0]

And loop only from 1 to length of array -1 and check the same condition as before.

Condensed Solution:

<https://gist.github.com/vermaayush680/7d3c7088697066de373be4f825ce42f5>

```
c=0
f=[0]+f+[0]
for i in range(1,len(f)-1):
    if f[i-1]==f[i]==f[i+1]==0:
        c+=1
        f[i]=1
return c>=n
```