Описание алгоритма

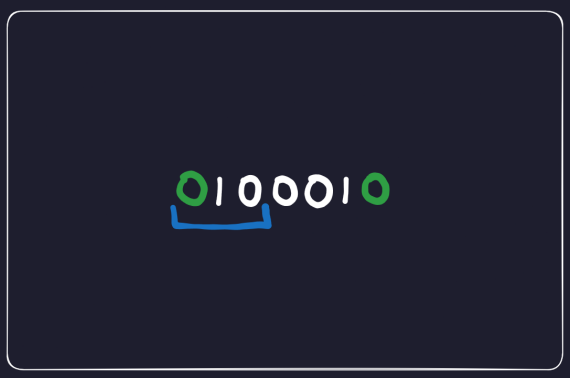
Алгоритм принимает на вход массив *flowerbed*, где 0 обозначает пустое место, а 1 — посаженный цветок, и число n — количество цветов, которое нужно посадить. Цветы нельзя сажать рядом.

Алгоритм работает следующим образом:

1. **Добавление нулей:** В начало и конец массива *flowerbed* добавляются нули, чтобы упростить обработку краевых случаев.
2. **Скользящее окно:** По массиву перемещается окно фиксированного размера.
3. **Проверка:** На каждой итерации проверяется, есть ли в текущем окне единица.
4. **Посадка:** Если единицы нет, алгоритм уменьшает счетчик n и "сажает"цветок в середину окна (устанавливает элемент массива в 1).
5. **Сдвиг:** Сдвигает окно на одну ячейку вправо.
6. **Результат:** Алгоритм возвращает True, если n становится равным 0, и False в противном случае.

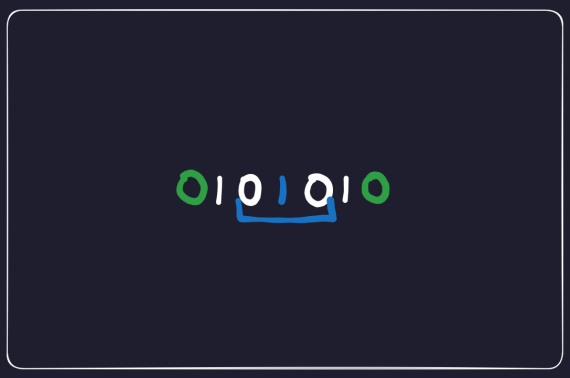
Визуализация

(a) Добавление нулей



(b) Окно на первых трёх элементах

(c) Сдвиг окна до трёх нулей



(d) Замена центрального нуля на 1

Рис. 1: Этапы обработки массива с использованием скользящего окна

Листинг

```
public bool Flowerbed(List<int> flowerbed, int n)
{
    flowerbed.Insert(0, 0);
    flowerbed.Add(0);

    int start = 0;
    int end = 3;

    while (n > 0 && end < flowerbed.Count)
    {
        List<int> subflower = flowerbed.GetRange(start, end - start);
        if (!subflower.Contains(1))
        {
            n--;
            flowerbed[(end + start) / 2] = 1;
        }
        start++;
        end++;
    }

    return n == 0;
}
```

Листинг 1: Csharp