

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 7

«Жадные алгоритмы»

Выполнил работу

Карташов Игорь

Академическая группа №J3111

Принято

Вершинин Владислав

Санкт-Петербург

2024

Введение

Цель задачи — вычислить минимальное количество конфет, которое нужно раздать детям, стоящим в линии, при соблюдении следующих условий:

- Каждый ребёнок должен получить как минимум одну конфету.
- Ребёнок с более высоким рейтингом должен получить больше конфет, чем его соседи с более низким рейтингом.

Теоретическая подготовка

Жадный алгоритм нужен, потому что:

1. Мы решаем задачу шаг за шагом, учитывая только соседей каждого ребёнка.
2. Локально правильные решения (например, дать больше конфет ребёнку с более высоким рейтингом) ведут к общему минимальному числу конфет.
3. Такой подход простой и быстрый — два прохода по массиву, без лишних вычислений.

Реализация

Шаги алгоритма:

1. Инициализация:

Создаётся массив `candies`, где каждому ребёнку изначально даётся одна конфета.

2. Первый проход (слева направо):

Если текущий ребёнок имеет рейтинг больше, чем у левого соседа, то он получает на одну конфету больше, чем сосед.

3. Второй проход (справа налево):

Если текущий ребёнок имеет рейтинг больше, чем у правого соседа, то он получает максимум из уже рассчитанного значения и количества конфет правого соседа плюс одна.

4. Подсчёт результата:

Суммируются все элементы массива `candies`, чтобы получить общее количество конфет.

Время: $O(n)$, так как два линейных прохода.

Память: $O(n)$, из-за использования массива `candies`.

Тестовая часть

Greedy

Run

Submit

0

Premium

Description

Editorial

Solutions

Submissions

Status	Language	Runtime	Memory	Notes
2 Accepted a minute ago	C++	0 ms	21.7 MB	
1 Accepted 7 minutes ago	C++	0 ms	21.8 MB	

Accepted

48 / 48 testcases passed

Editorial

Solution

user2222Qx submitted at Dec 27, 2024 21:34

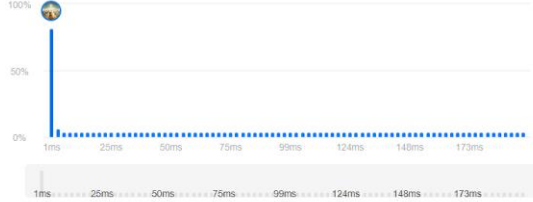
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

21.67 MB | Beats 35.71%



Code: C++

```
class Solution {
public:
    int candy(std::vector<int>& ratings) {
```

Testcase

Test Result

Заключение

Жадный алгоритм подходит для этой задачи, потому что он быстрый, простой и сразу минимизирует количество конфет, соблюдая правила.

Приложения

```
class Solution {
public:
    int candy(std::vector<int>& ratings) {
        int n = ratings.size(); // O(1)
        std::vector<int> candies(n, 1); // O(n), 4*n bytes

        //проход слева направо
        for (int i = 1; i < n; ++i) { // O(n)
            if (ratings[i] > ratings[i - 1]) {
                candies[i] = candies[i - 1] + 1; //O(1)
            }
        }

        //проход справа налево
        for (int i = n - 2; i >= 0; --i) { // O(n)
            if (ratings[i] > ratings[i + 1]) {
                candies[i] = std::max(candies[i], candies[i + 1] + 1); // O(1)
            }
        }

        // Подсчёт общей суммы конфет
        int total = 0;
        for (int candy : candies) { //O(n)
            total += candy;
        }

        return total;
    }
};

// Итого: O(n)
// Общая память: O(n) (основной вклад вносят массив candies)
```