ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 6

«Super egg drop»

Выполнил работу

Фамилия Имя

Семенов Никита

Академическая группа №группы

j3114, 467414

Принято

Должность, звание Фамилия Имя преподавателя

Санкт-Петербург

2024

**Задача** 887 [**Super Egg Drop**](https://leetcode.com/problems/super-egg-drop/) с leetcode

**Краткое описание задачи**

Дано k и n – кол-во яиц и кол-во этажей соответственно.

Яйцо, падая с этажа может либо разбиться, либо нет.

Необходимо найти минимальное кол-во бросков (m), за которое можно точно определить максимальный этаж, падая с которого, яйцо не разобьётся.

**Код решения**

#include <iostream>

#include <vector>

using namespace std;

class Solution {

public:

    int superEggDrop(int k, int n) {

        vector<vector<int>> dp(k + 1, vector<int>(n + 1, 0));

        int m = 0;

        while (dp[k][m] < n) {

            m++;

            for (int i = 1; i <= k; i++) {

                dp[i][m] = dp[i - 1][m - 1] + dp[i][m - 1] + 1;

            }

        }

        return m;

    }

}

**Краткое описание решения**

Создаётся табличка k+1 на n+1 элементов, заполненная нулями.

k – кол-во яиц

m – кол-во бросков

Далее разбиваем задачу на 3 случая – этажи выше, ниже и наш этаж.

Получаем формулу: dp[i][m] = dp[i - 1][m - 1] + dp[i][m - 1] + 1

Повторяем, увеличивая m, пока dp[i][m] меньше кол-ва этажей.

dp[i][m] – кол-во этажей, которое мы можем проверить за m бросков.

**Подсчёт асимптотики**

Максимальное число бросков m ограничено log(n), так как с каждым броском количество проверяемых этажей растёт экспоненциально.

На каждом броске m вычисляем значения для всех k яиц.

Значит сложность - O(k\*log(n))

k – кол-во яиц

m – кол-во бросков

**Пространственная сложность**

Создаётся таблица k+1 на n+1, тогда: O((k+1)⋅(n+1)) ∼ O(k⋅n)

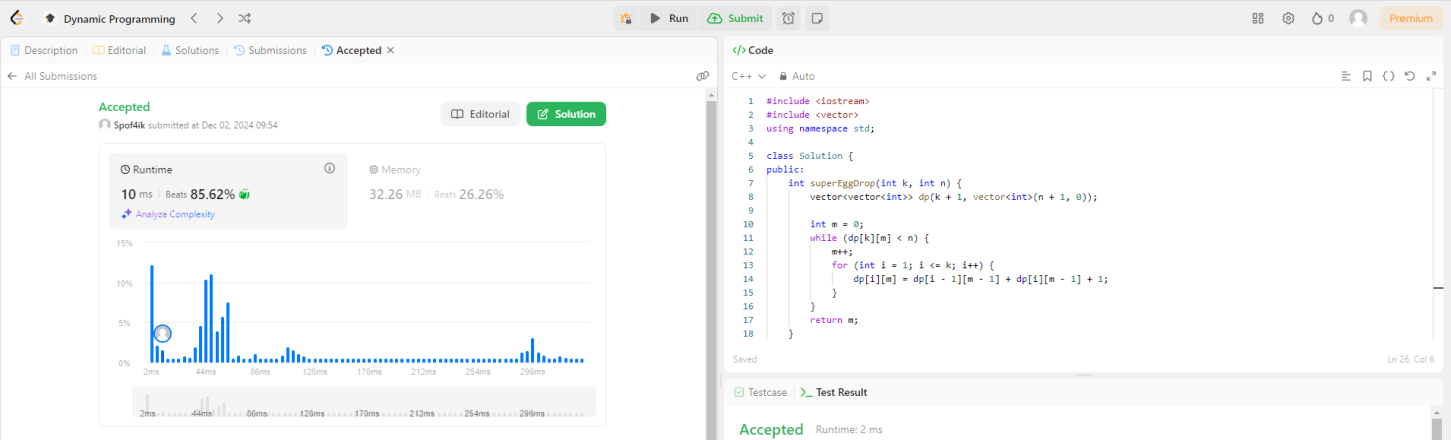
**Зачем использовать именно ДП в данной задаче?**

В данной задаче есть множество **повторяющихся подзадач**, результаты которых можно сохранить и использовать в дальнейшем решении. Поэтому использование ДП облегчает понимание задачи. Также ДП даёт быстрое и удобное решение, обеспечивая баланс между скоростью выполнения и простотой реализации.

**Что можно улучшить?**

Можно создавать таблицу не k+1 на n+1 элементов, а k на n. Тогда станет чуть тяжелее построить нужный алгоритм (подобрать индексы) и придётся прописывать некоторые случаи вручную.

**Скрин с решением на leetcode**

****