



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

"МИРЭА - Российский технологический университет"

РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра МОиСИТ

**ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ №8.2**

**«Реализация алгоритмов на основе сокращения числа переборов»
по дисциплине
«Структуры и алгоритмы обработки данных»**

Выполнил студент группы ИКБО-10-24

Таганов А.А.

Москва 2025

Цель работы: разработать алгоритм решения задачи с применением метода, указанного в варианте и реализовать программу.

Задача: Дано прямоугольное поле размером $n \times m$ клеток. Можно совершать шаги длиной в одну клетку вправо, вниз или по диагонали вправо-вниз. В каждой клетке записано некоторое натуральное число. Необходимо попасть из верхней левой клетки в правую нижнюю. Вес маршрута – это сумма чисел всех посещенных клеток. Найти маршрут с минимальным весом.

Метод: Динамическое программирование.

Решение в лоб заключается в следующем. Наша функция будет рекурсивно вызвана по трем направлениям. Затем будет выбран наименьший путь.

Листинг 1 – Решение “в лоб”

```
long long bruteRecursive(int x, int y,
                        const vector<vector<int>>& a,
                        int n, int m)
{
    if (x >= n || y >= m)
        return INF;

    if (x == n - 1 && y == m - 1)
        return a[x][y];

    bruteOperationsCount += 3;

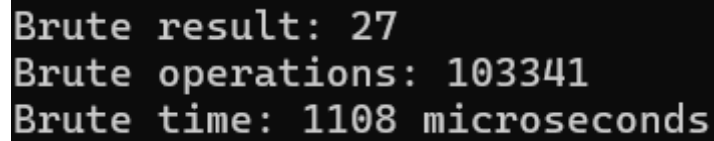
    long long right  = bruteRecursive(x,      y + 1, a, n, m);
    long long down   = bruteRecursive(x + 1, y,   a, n, m);
    long long diag   = bruteRecursive(x + 1, y + 1, a, n, m);

    long long best = min({right, down, diag});

    if (best == INF) return INF;

    return a[x][y] + best;
}
```

Результаты тестирования представлены на рисунке 1.



```
Brute result: 27
Brute operations: 103341
Brute time: 1108 microseconds
```

Рисунок 1 – Результат тестирования программы

Для эффективного решения составим матрицу $n \times m$, заполним ее нулями, заполним первую строку и первый столбец, а затем будем высчитывать минимальный клетку из которой можем пойти в новую клетку, и таким образом составим матрицу, где в правой нижней клетке будет храниться минимальный вес пути.

Листинг 2 – Эффективное решение

```
long long solveDP(const vector<vector<int>>& a, int n, int m)
{
    vector<vector<long long>> dp(n, vector<long long>(m, 0));

    dp[0][0] = a[0][0];
    dpOperationsCount++;

    for (int j = 1; j < m; ++j) {
        dp[0][j] = dp[0][j - 1] + a[0][j];
        dpOperationsCount += 2;
    }

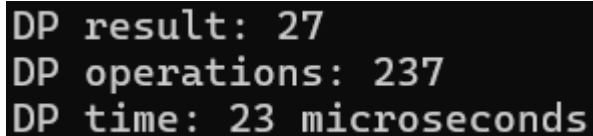
    for (int i = 1; i < n; ++i) {
        dp[i][0] = dp[i - 1][0] + a[i][0];
        dpOperationsCount += 2;
    }

    for (int i = 1; i < n; ++i) {
        for (int j = 1; j < m; ++j) {
            long long bestPrev = min({
                dp[i - 1][j],
                dp[i][j - 1],
                dp[i - 1][j - 1]
            });

            dp[i][j] = a[i][j] + bestPrev;
            dpOperationsCount += 5;
        }
    }
}
```

```
}  
  
return dp[n - 1][m - 1];  
}
```

Результат тестирования представлены на рисунке 2.



```
DP result: 27  
DP operations: 237  
DP time: 23 microseconds
```

Рисунок 2 – Результаты тестирования программы

Вывод

Поставленные задачи выполнены: реализованы алгоритмы решения поставленных задач. Реализовано решения «в лоб», с помощью рекурсии, и эффективное решения, использующее методы динамического программирования.

Литература.

1. Страуструп Б. Программирование. Принципы и практика с использованием C++. 2-е изд., 2016.
2. Документация по языку C++ [Электронный ресурс]. URL: <https://docs.microsoft.com/ruru/cpp/cpp/> (дата обращения 04.12.2025).
3. Курс: Структуры и алгоритмы обработки данных. Часть 2 [Электронный ресурс]. URL: <https://online-edu.mirea.ru/course/view.php?id=4020> (дата обращения 04.12.2025).