МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА» - Российский технологический университет»

РТУ МИРЭА

Отчёт по выполнению практического задания № 8.2

Тема:

«Реализация алгоритмов на основе сокращения числа переборов»

Выполнил студент: Данов Арсений Иванович

Группа: ИКБО-10-23

Москва 2024

Вариант:

Номер: *6*

Задача:

Дано прямоугольное поле размером n*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.

1.1 Формулировка задачи

1. Разработать алгоритм решения задачи с применением метода, указанного в варианте и реализовать программу.
2. Оценить количество переборов при решении задачи стратегией «в лоб» - грубой силы. Сравнить с числом переборов при применении метода.
3. Оформить отчет в соответствии с требованиями документирования разработки ПО.

1.2 Ход решения

Для переборного решения необходимо проверить все возможные пути из левой верхней клетки в правую нижнюю, чьё число возрастает очень быстро.

Для эффективного решения достаточно лишь каждый раз для каждой клетки выбирать из трех возможных чисел, откуда можно в неё попасть, наименьшее и прибавлять к нему собственное значение клетки.

Таким образом после применения алгоритма ко всем клеткам в после завершения работы алгоритма получим в правой нижней клетке минимальное возможное число.

1.3 Исходный код программ

На рисунках 1-2 представлены исходный коды функций нахождения количества возможных вариантов и нахождения минимальной суммы.

```
8   vector<vector<int>> count_variants(int n, int m)
9   {
10      vector<vector<int>> arr(n, vector<int>(m, 0));
11      arr[0][0] = 1;
12      for (int i = 1; i < n; ++i)
13      {
14          arr[i][0] += arr[i - 1][0];
15      }
16      for (int j = 1; j < m; ++j)
17      {
18          arr[0][j] += arr[0][j - 1];
19      }
20      for (int i = 1; i < n; ++i)
21      {
22          for (int j = 1; j < m; ++j)
23          {
24              arr[i][j] += arr[i - 1][j] + arr[i][j - 1] + arr[i - 1][j - 1];
25          }
26      }
27      return arr;
28  }
```

Рисунок 1 — Функция нахождения количества путей

```cpp
vector<vector<int>> min_sum(vector<vector<int>> &base_arr)
{
    int n = base_arr.size();
    int m = base_arr[0].size();
    vector<vector<int>> arr(n, vector<int>(m, 0));
    arr[0][0] = base_arr[0][0];
    for (int i = 1; i < n; ++i)
    {
        arr[i][0] = arr[i - 1][0] + base_arr[i][0];
    }
    for (int j = 1; j < m; ++j)
    {
        arr[0][j] = arr[0][j - 1] + base_arr[0][j];
    }
    for (int i = 1; i < n; ++i)
    {
        for (int j = 1; j < m; ++j)
        {
            arr[i][j] = base_arr[i][j] + min(arr[i - 1][j], arr[i][j - 1], arr[i - 1][j - 1]);
        }
    }
    return arr;
}
```

Рисунок 2 — Функция нахождения минимальной суммы

1.4 Результаты тестирования

На рисунках 3-4 представлены результаты тестирования программы.

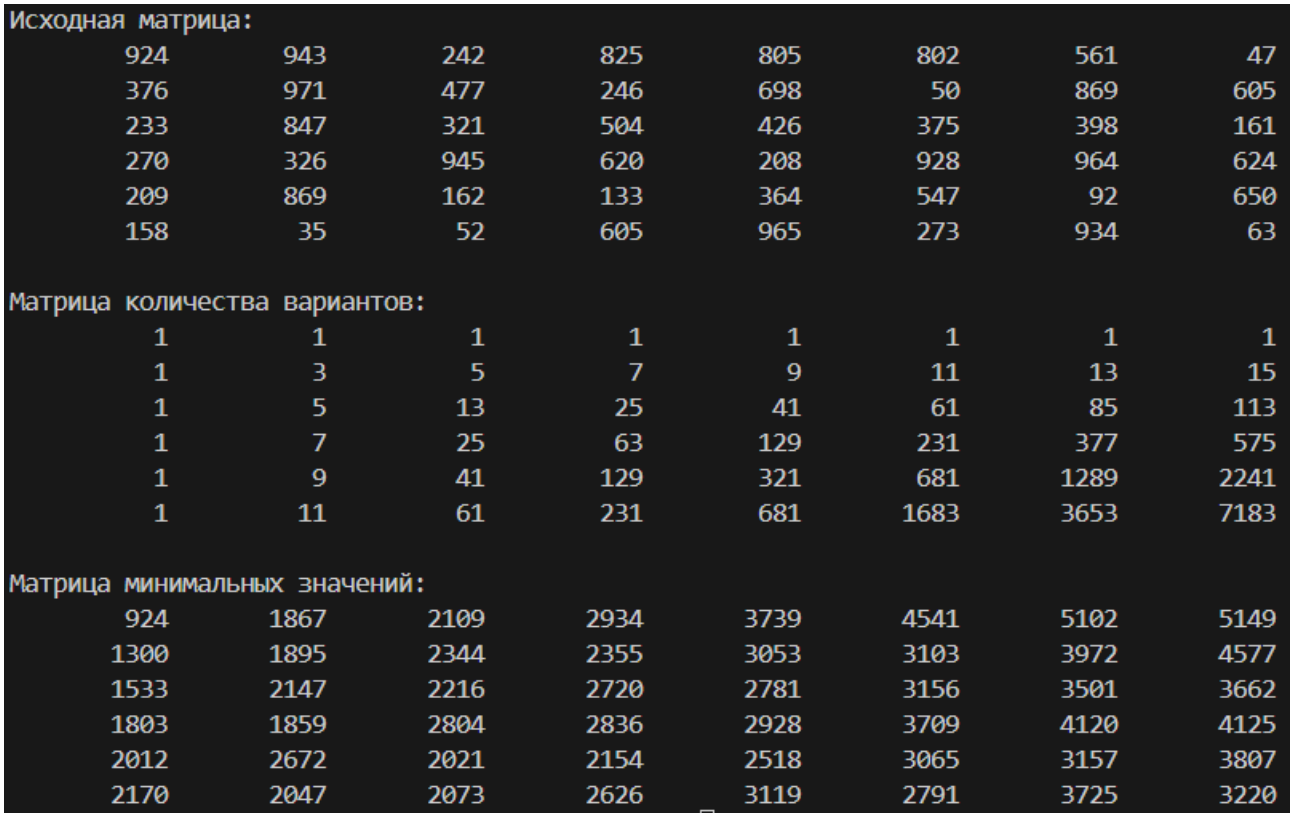

Рисунок 7 — Результаты тестирования программы

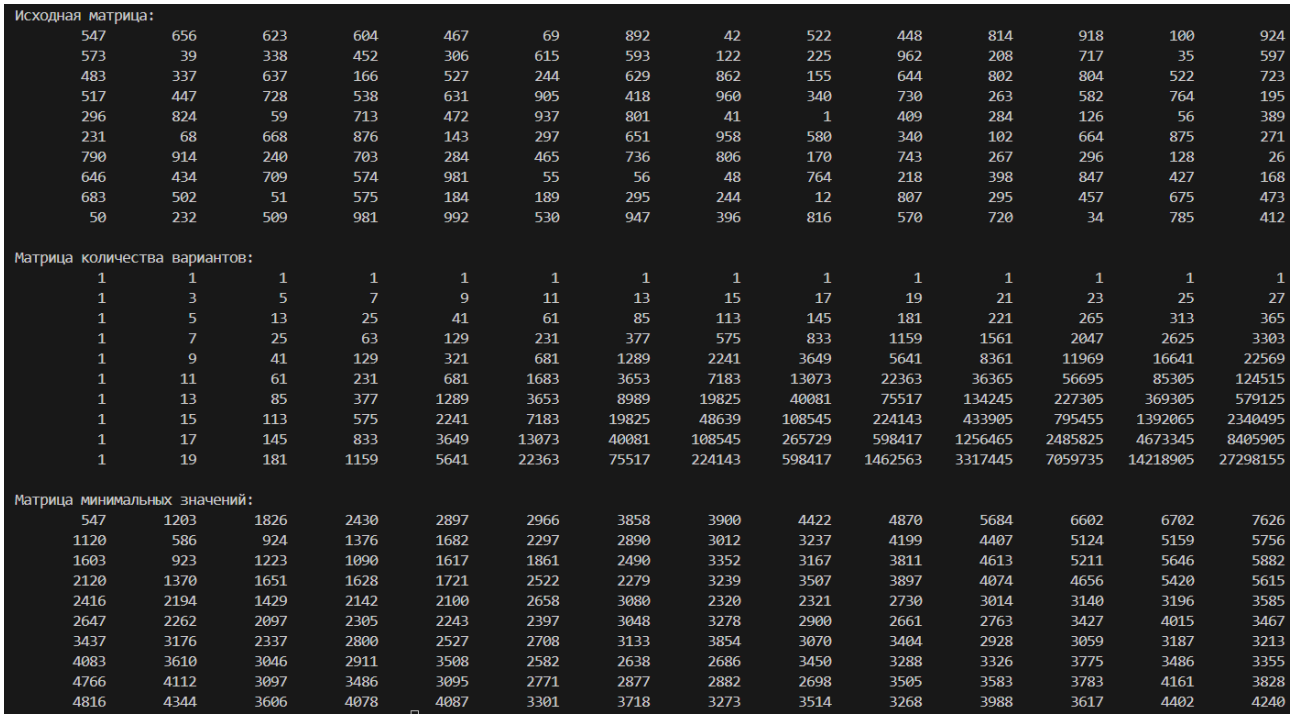

Рисунок 8 — Результаты тестирования программы

2.1 Выводы

Для большинства задач помимо полного переборного решения существуют также и эффективные алгоритмы, позволяющие во много раз сократить сложность задачи и время решения.