Министерство образования и науки РФ

Новосибирский государственный технический университет

Кафедра систем сбора и обработки данных

**Лабораторная работа №4**

Вариант 15.

Динамическое программирование

Факультет: АВТФ Преподаватель: Квашнина Е. А.

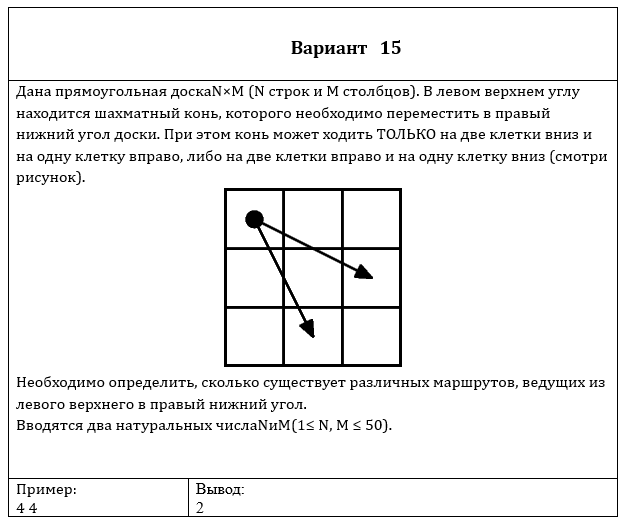
Группа: АТ-33

Студент: Мельничук В.

Новосибирск, 2025

1. Формулировка задания

Разработайте решение задачи в соответствии с вариантом с помощью методов динамического программирования.



1. Описание принципа

В задаче применяется принцип динамического программирования, при котором:

* создаётся двумерный массив dp[N][M], где dp[i][j] — количество способов попасть в клетку (i, j);
* начальная клетка (0, 0) инициализируется единицей: туда можно попасть только одним способом — начав путь;
* каждая клетка (i, j) заполняется суммой значений из двух клеток, из которых в неё может попасть конь:
  + из (i−2, j−1)
  + из (i−1, j−2);
* таким образом, каждая ячейка зависит от уже вычисленных значений, что исключает повторные расчёты.

Алгоритм работает за O(N\*M) и использует O(N\*M) памяти.

1. Листинг программы

**main.cpp**

#include <iostream>

#include <vector>

// Функция считает количество путей шахматного коня

int count\_knight\_paths(int N, int M) {

// Инициализация таблицы DP нулями

    std::vector<std::vector<int>> dp(N, std::vector<int>(M, 0));

// Стартовая точка (0, 0) — один способ попасть

    dp[0][0] = 1;

// Обход всех клеток доски

    for (int i = 0; i < N; ++i) {

        for (int j = 0; j < M; ++j) {

// Если возможно, добавить путь от клетки (i-2, j-1)

            if (i - 2 >= 0 && j - 1 >= 0)

                dp[i][j] += dp[i - 2][j - 1];

// Если возможно, добавить путь от клетки (i-1, j-2)

            if (i - 1 >= 0 && j - 2 >= 0)

                dp[i][j] += dp[i - 1][j - 2];

        }

    }

// Возвращаем количество способов попасть в правый нижний угол

    return dp[N - 1][M - 1];

}

int main() {

    int N, M;

    std::cin >> N >> M;

    std::cout << count\_knight\_paths(N, M) << std::endl;

    return 0;

}