

26. Самый дешевый путь

Ограничение времени	1 секунда
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В каждой клетке прямоугольной таблицы $N \times M$ записано некоторое число. Изначально игрок находится в левой верхней клетке. За один ход ему разрешается перемещаться в соседнюю клетку либо вправо, либо вниз (влево и вверх перемещаться запрещено). При проходе через клетку с игрока берут столько килограммов еды, какое число записано в этой клетке (еду берут также за первую и последнюю клетки его пути).
Требуется найти минимальный вес еды в килограммах, отдав которую игрок может попасть в правый нижний угол.

Формат ввода

Вводятся два числа N и M — размеры таблицы ($1 \leq N \leq 20, 1 \leq M \leq 20$). Затем идет N строк по M чисел в каждой — размеры штрафов в килограммах за прохождение через соответствующие клетки (числа от 0 до 100).

Формат вывода

Выведите минимальный вес еды в килограммах, отдав которую можно попасть в правый нижний угол.

Пример

Ввод	Вывод
5 5 1 1 1 1 1 3 100 100 100 100 1 1 1 1 1 2 2 2 2 1 1 1 1 1 1	11

Язык

GNU GCC 12.2 C++20

Набрать здесь

Отправить файл

```
1 #include <iostream>
2 #include <vector>
3
4 int main() {
5     int n, m;
6     std::cin >> n >> m;
7     std::vector<std::vector<int>> dp(n);
8     std::vector<std::vector<int>> in(n);
9     for (int i = 0; i != n; ++i) {
10         dp[i].resize(m, 40400);
11         in[i].resize(m);
12     }
13     for (int i = 0; i != n; ++i) {
14         for (int j = 0; j != m; ++j) {
15             std::cin >> in[i][j];
16         }
17     }
18     dp[0][0] = in[0][0];
19     for (int i = 0; i != n; ++i) {
20         for (int j = 0; j != m; ++j) {
21             if (i > 0) {
22                 dp[i][j] = std::min(dp[i][j], dp[i - 1][j] + in[i][j]);
23             }
24             if (j > 0) {
25                 dp[i][j] = std::min(dp[i][j], dp[i][j - 1] + in[i][j]);
26             }
27         }
28     }
29     std::cout << dp[n - 1][m - 1];
30     return 0;
31 }
```

Отправить