Московский авиационный институт
(национальный исследовательский университет)

Институт информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №7 по курсу дискретного анализа

Студент: А. В. Синявский
Преподаватель: Н. А. Зацепин
Группа: М8О-308Б-18
Дата:
Оценка:
Подпись:

Москва, 2020

Лабораторная работа № 7 по курсу дискретного анализа

Выполнил студент группы М80-308Б-18 МАИ *Синявский Андрей.*

Условие

При помощи метода динамического программирования разработать алгоритм решения задачи, определяемой своим вариантом
Разработать программу на языке C или C++, реализующую построенный алгоритм.

Вариант №2

Задан прямоугольник с высотой n и шириной m, состоящий из нулей и единиц. Найдите в нём прямоугольник наибольшой площади, состоящий из одних нулей.

Метод решения

Для каждой строки матрицы динамически составим вектор длины m, j-ая позиция которого будет содержать длину последовательности нулей в j-ом столбце, оканчивающейся в этой позиции (т.е. если A[i,j]=1, то i-ый вектор в позиции j содержит ноль, иначе значение j-ой позиции вектора i-1, увеличенное на 1). При помощи полученных векторов для каждой позиции в векторе определим максимальный размер искомого прямоугольника с высотой, равной значению рассматриваемой позиции. Максимум по всем найденным размерам даст нам ответ.

Описание программы

Так как весь код программы написан внутри блока main, и ни классов, ни функций для описания не имеет, а табличку чем-то заполнить надо, опишу блоки кода с указанием строк начала и конца

Блок	Предназначение
6-7	оптимизация ввода-вывода
9-19	ввод размеров матрицы и выделение памяти под вектора, необходимые для описанного решения
21-32	ввод строк матрицы, и одновременно динамическое заполнение массивов (явно хранение матрицы в программе не предусмотрено, по входным данным сразу вычисляется динамика)
34-48	вычисление правых границ прямоугольников
50-64	вычисление левых границ прямоугольников
99-73	нахождение максимальной площади

Дневник отладки

При создании этой таблицы была использована история посылок.

Время	ведрикт	Описание
2020/10/03 09:02:07	Ошибка выполнения	выход за границы массива
2020/10/03 10:46:102	Ожидает подтверждения	ура

листинг

main.cpp

```
#include <iostream>
#include <stack>
#include <vector>

int main() {
    std::ios::sync_with_stdio(false);
    std::cin.tie(nullptr);

    int n, m;
    std::cin >> n >> m;
```

```
std::vector<std::vector<int>> matrix;
matrix.resize(n);
for (auto &i : matrix) i.resize(m);
std::vector<std::vector<int>> R;
R.resize(n);
for (auto &i : R) i.resize(m);
std::vector<std::vector<int>> L;
L.resize(n);
for (auto &i : L) i.resize(m);

char c;
for (int i = 0; i < m; ++i) {
    std::cin >> c;
    matrix[0][i] = (c == '0');
}
for (int i = 1; i < n; ++i) {
    for (int j = 0; j < m; ++j) {
        std::cin >> c;
        if (c == '1') matrix[i][j] = 0;
        else matrix[i][j] = matrix[i-1][j] + 1;
    }
}

for (int i = 0; i < n; ++i) {
    std::stack<int> S;
    for (int j = 0; j < m; ++j) {
        if (!S.empty()) while (matrix[i][j] < matrix[i][S.top()]) {
                R[i][S.top()] = j-1;
                S.pop();
                if (S.empty()) break;
            }
        S.push(j);
    }
    while (!S.empty()) {
        R[i][S.top()] = m - 1;
        S.pop();
    }
}

for (int i = 0; i < n; ++i) {
    std::stack<int> S;
    for (int j = m-1; j >= 0; --j) {
```

```
            if (!S.empty()) while (matrix[i][j] < matrix[i][S.top()]) {
                    L[i][S.top()] = j+1;
                    S.pop();
                    if (S.empty()) break;
                }
            S.push(j);
        }
        while (!S.empty()) {
            L[i][S.top()] = 0;
            S.pop();
        }
    }

    int max = 0;
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            int tmp = matrix[i][j]*(R[i][j] - L[i][j] + 1);
            if (tmp > max) max = tmp;
        }
    }
    std::cout << max << '\n';

    return 0;
}
```

Недочёты

Посимвольный ввод. Можно было бы считывать матрицу быстрее, но пришлось бы парсить строки.

Выводы

Проделав данную работу, я изучил метод динамического программирования, что, я надеюсь, поможет мне писать оптимальный код в будущем.