МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ УПРАВЛІННЯ

ЗВІТ

З ЛАБОРАТОРНОЇ РОБОТИ №8

ЗА ТЕМОЮ “ ДИНАМІЧНЕ ПРОГРАМУВАННЯ”

Виконав студент

групи КН-221д

Габорець

Перевірив

Солонська С.В.

Харків 2022

ЛАБОРАТОРНА РОБОТА №8

Тема: ДИНАМІЧНЕ ПРОГРАМУВАННЯ

Мета роботи: навчитися використовувати динамічне програмування та оцінювати його складність.

Завдання :

Розробити програму, яка читає з клавіатури вхідні дані та розв’язує задачу методом динамічного програмування. Визначити складність алгоритму

Варіант завдань

1 Пошук маршруту на прямокутному полі так, щоб сума чисел у клітинах, через які він іде, була максимальною. Рух починається із клітинки верхнього лівого кута та завершується у правому нижньому. Кожний крок виконується на одну позицію праворуч або вниз. Вхідні дані: натуральні числа N, M (1< N, M < 256) та послідовність N \* M натуральних чисел — прямокутне поле рядок за рядком. Вихідні дані: таблиця динамічного програмування (A(i,j) = максимальна сума маршруту до клітинки (i, j)); прямокутне поле із маршрутом, що відмічено зірочками, та сума чисел у клітинках маршруту.

Сложность алгоритма O(n^2)-Два вложенных цикла



КОД:

#include <vector>

#include<iostream>

#include <algorithm>

using namespace std;

void print(const vector<vector<int>>& vector)

{

cout << endl;

for (auto& i : vector) {

for (auto& j : i) {

cout << j<<"\t";

}

cout << endl;

}

}

//#define int long long int

int gcd(int a, int b) { return gcd(a, b); }

int lcm(int a, int b) { return abs(a \* b) / gcd(a, b); }

int main() {

//ios\_base::sync\_with\_stdio(false);

//cin.tie(0);

//cout.tie(0);

/\* cout.precision(9);\*/

/\*cout << fixed;\*/

cout << "Enter n,m(1<n,m<256)\n";

int n;

int m;

cin >> n >> m;

vector<vector<int>> a(n, vector<int>(m));

cout << "Enter array`s data\n";

for (auto& i : a) {

for (auto& j : i) {

char c;

cin >> c;

j = c - '0';

}

}

print(a);

vector<vector<int>> dp(n, vector<int>(m));

dp[0][0] = a[0][0];

for (int i = 0; i < n; ++i) {//Сложность 0(n^2)

for (int j = 0; j < m; ++j) {

if (i == 0 && j == 0)

continue;

if (i == 0) {

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

}

else if (j == 0) {

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

}

else {

dp[i][j] = a[i][j] + max(dp[i][j - 1], dp[i - 1][j]);

}

}

}

print(dp);

vector<vector<char>> ans(n, vector<char>(m, '-'));

vector<int> dx = { 0, -1 };

vector<int> dy = { -1, 0 };

int x = n - 1;

int y = n - 1;

ans[x][y] = '\*';

while (x != 0 || y != 0) {

int max\_one = numeric\_limits<int>::min();

int max\_i = 0;

int max\_j = 0;

for (size\_t d = 0; d < dx.size(); ++d) {

int i = x + dx[d];

int j = y + dy[d];

bool ok = i >= 0 && j >= 0 && i < n&& j < n;

if (ok && max\_one < dp[i][j]) {

max\_one = dp[i][j];

max\_i = i;

max\_j = j;

}

}

ans[max\_i][max\_j] = '\*';

x = max\_i;

y = max\_j;

}

cout << dp[n - 1][m - 1] << '\n';

ans[0][0] = '\*';

for (const auto& i : ans) {

copy(i.cbegin(), i.cend(), ostream\_iterator<char>(cout, ""));//вывести

cout << '\n';

}

}

Висновок: виконавши лабораторну роботу №8, я навчився використовувати динамічне програмування та оцінювати його складність.