

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ
ТЕХНІЧНИЙ УНІВЕРСИТЕТ «ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ
ІНСТИТУТ»

Кафедра «Програмна інженерія та інформаційні технології управління»

Звіт з лабораторної роботи №8

З предмету «Алгоритми та структури даних»

Виконав Студент групи КН-36а

Рубан Ю.Д.

Перевірила:

ас. Бородіна І. О.

Харків 2017

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 завдання:

Пошук найбільшої спільної підпослідовності. Вхідні дані: натуральні числа N, M ($1 < N, M < 256$) та дві послідовності X та Y натуральних чисел довжиною N та M відповідно. Вихідні дані: динамічна таблиця ($A(i,j)$ = довжина НСП для префіксів X_{i-1} та Y_j) та НСП для X та Y .

Хід виконання роботи

Розроблено програму, яка читає з клавіатури вхідні дані та розв'язує задачу методом динамічного програмування. Визначено складність алгоритму.

Код програми:

```
#include<iostream>
using namespace std;

int max(const int&a, const int&b);
int** getRes(int **res, int*x, int*y, int n, int m);
int* getSubSequence(int **res, int*x, int*y, int n, int m);
int P=0;
int main()
{
    int n, m;
    cout << "enter N, M" << endl;
    cin >> n >> m;
    int *X = new int[n];
    int *Y = new int[m];
    cout << "enter first sequence" << endl;
    for (int i = 0; i < n; i++)
    {
        cin >> X[i];
    }
    cout << "enter second sequence" << endl;
    for (int i = 0; i < m; i++)
    {
        cin >> Y[i];
    }
    int* l = new int[max(m,n)];
    int **res = new int*[n+1];
    for(int i=0; i<n+1; i++)
    {
        res[i] = new int[m+1];
    }
    for (int i = 0; i < n+1; i++)
    {
        for (int j = 0; j < m+1; j++)
        {
            res[i][j] = 0;
        }
    }
    res = getRes(res, X, Y, n, m);
    cout << "length = " << res[0][0] << endl;
    cout << "dynamic table: " << endl;
```

```

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            cout << res[i][j]<<" ";
        }
        cout << endl;
    }
    int *sub = getSubSequence(res, X, Y, n, m);
    cout << "subSequence:" << endl;
    for (int i = 0; i < res[0][0]; i++)
    {
        cout << sub[i] << " ";
    }
    cout << endl;
    cout << "algorithm complexity= " << P << endl;
    cout << endl;
    system("pause");
    return 0;
}

int max(const int&a,const int&b) { return a > b ? a : b; }
int** getRes(int **res,int*x, int*y, int n, int m)
{
    for (int i = n - 1; i >= 0; i--)
    {
        for (int j = m - 1; j >= 0; j--)
        {
            if (x[i] == 0 || y[j] == 0) { res[i][j] = 0; }
            else if (x[i] == y[j]) { res[i][j] = res[i + 1][j + 1] + 1; }
            else
            {
                res[i][j] = max(res[i + 1][j], res[i][j + 1]);
            }
            P++;
        }
    }
    return res;
}

int* getSubSequence(int **res,int*x,int*y, int n, int m)
{
    int *sub = new int[res[0][0]];
    int i = 0;
    int j = 0;
    int c = 0;
    while (i<n &&j<m)
    {
        if (x[i] == y[j])
        {
            sub[c] = x[i];
            c++; i++; j++;
        }
        else if (res[i + 1][j] > res[i][j + 1]) { i++; }
        else { j++; }
        P++;
    }
    return sub;
}

```

Висновки

У даній лабораторній роботі я познайомився з принципами динамічного програмування та навчився визначати складність алгоритму.