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 і структури даних»

Виконав:

ст. гр. КН-36а

Кулик В.В

Перевірили:

ас. каф. ПІІТУ

І.О. Бородіна

Харків 2017

**Мета роботи:**

Навчитися використовувати динамічне програмування та оцінювати його складність.

**Завдання:**

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

**Варіант 2.**

**Хід роботи:**

#include<iostream>

#include<string>

#include<vector>

using namespace std;

int max(const int&a, const int&b);

int\*\* getRes(int \*\*res, string x, string y);

string getSubSequence(int \*\*res, string x, string y);

void getArrows(string x, string y);

int main()

{

setlocale(0, "Russian");

string X;

string Y;

cout << "Enter sequences: " << endl;

cin >> X;

cin >> Y;

int n = X.size();

int m = Y.size();

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;

}

}

cout << "Dynamic table:" << endl;

res = getRes(res, Y, X);

string sub = getSubSequence(res, Y, X);

cout << "Arrows: " << endl;

getArrows(Y, X);

cout << "Subsequence:" << endl;

for (int i = 0; i < sub.size(); i++)

{

cout << sub[i] << " ";

}

cout << endl;

system("pause");

return 0;

}

int max(const int&a, const int&b) { return a > b ? a : b; }

int\*\* getRes(int \*\*res, string x, string y)

{

int n = x.size();

int m = y.size();

cout << " ";

for (int i = 0; i<y.size(); i++)

{

cout << " " << y[i];

}

cout << endl;

for (int i = 1; i <= n; i++)

{

cout << x[i - 1] << " ";

for (int j = 1; j <= m; j++)

{

if (x[i - 1] == '\0' || y[j - 1] == '\0') { res[i][j] = 0; }

else if (x[i - 1] == y[j - 1]) { res[i][j] = res[i - 1][j - 1] + 1; }

else

{

res[i][j] = max(res[i - 1][j], res[i][j - 1]);

}

cout << res[i][j] << " ";

}

cout << endl;

}

return res;

}

string getSubSequence(int \*\*res, string x, string y)

{

int n = x.size();

int m = y.size();

int size = res[m][n];

int i = x.size() - 1;

int j = y.size() - 1;

string sub;

int ik, jk;

ik = i + 1;

jk = j + 1;

while (i >= 0 && j >= 0)

{

if (x[i] == y[j])

{

sub += x[i];

i--; j--;

ik--;

jk--;

}

else if (res[ik - 1][jk] >= res[ik][jk - 1]) { i--; ik--; }

else { j--; jk--; }

}

string result;

for (int i = sub.size() - 1; i >= 0; i--)

{

result += sub[i];

}

return result;

}

void getArrows(string x, string y)

{

int n = x.size();

int m = y.size();

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;

}

}

vector<string>arrows(n + 1);

for (int i = 0; i < n + 1; i++)

{

arrows[i].resize(m + 1);

}

int ik = 0, jk = 0;

cout << " ";

for (int i = 0; i<y.size(); i++)

{

cout << " " << y[i];

}

cout << endl;

for (int i = 1; i <= n; i++)

{

cout << x[i - 1] << " ";

for (int j = 1; j <= m; j++)

{

if (x[i - 1] == '\0' || y[j - 1] == '\0') { res[i][j] = 0; }

else if (x[i - 1] == y[j - 1]) { res[i][j] = res[i - 1][j - 1] + 1; arrows[ik][jk] += 'd'; }

else if (res[i - 1][j] >= res[i][j - 1])

{

res[i][j] = res[i - 1][j];

arrows[ik][jk] += 'u';

}

else

{

res[i][j] = res[i][j - 1];

arrows[ik][jk] += 'l';

}

cout << arrows[ik][jk] << " ";

jk++;

}

ik++;

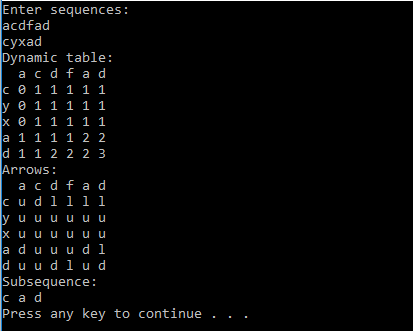
jk = 0;

cout << endl;

}

}

**Виконання:**

****

**Висновок:** в ході лабораторної роботи я ознайомився з принципами динамічного програмування.