Контрольная работа 3

Задание 1  
Код программы:  
#include <iostream>

using namespace std;

int\*\* M;

int\* Q, \* R, \* P, \* S;

int main()

{

int a, b, n;

cin >> n;

M = new int\* [n];

int m = n;

for (int i = 0; i < m; i++)

M[i] = new int[m];

int start, end, weight;

for (int i = 0; i < m; ++i)

{

cin >> start >> end >> weight;

M[start][end] = weight;

}

cin >> a;

cin >> b;

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

{

Q[i] = 1; R[i] = M[a][i]; P[i] = a;

}

Q[a] = 0; P[a] = 0;

int z, k, rmin;

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

{

rmin = 1000000;

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

if (Q[i] && (R[i] < rmin))

{

rmin = R[i]; k = i;

}

Q[k] = 0;

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

if (Q[i] && (z = M[k][i] + R[k]) < R[i])

{

R[i] = z; P[i] = k;

}

for (int j = 1; j < n && Q[b]; j++)

{

rmin = 1000000;

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

if (Q[i] && (R[i] < rmin))

{

rmin = R[i]; k = i;

}

Q[k] = 0;

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

if (Q[i] && (z = M[k][i] + R[k]) < R[i])

{

R[i] = z; P[i] = k;

}

}

int i;

i = P[b]; k = 0; S[0] = b;

while (i != P[a])

{

i = P[i]; k++; S[k] = i;

}

cout << R[b];

cout << "Road: ";

for ( ; k > 0; --k)

cout << S[k] << " ";

return 0;

}

}

Трудоемкость этого алгоритма для графа из n вершин и m ребер: O(m\*n)

Задание 2  
Код программы:

#include <iostream>

using namespace std;

int main()

{

int\* v1, \* v2, \* L, \* S, \* D, \* U, \* R, \* P;

int i, n, m, j, k, t, q;

scanf("%d%d", &n, &m);

v1 = new int[m]; v2 = new int[m];

for (i = 0; i < m; i++)

scanf("%d%d", &v1[i], &v2[i]);

D = new int[m]; S = new int[n];

L = new int[n]; U = new int[n];

for (j = 0; j < n; j++)

L[j] = 0;

for (i = 0; i < m; i++)

{

L[v1[i]]++;

}

S[0] = 0;

for (j = 1; j < n; j++)

S[j] = S[j - 1] + L[j - 1];

for (j = 0; j < n; j++)

U[j] = S[j];

for (i = 0; i < m; i++)

{

k = v1[i];

D[U[k]] = v2[i];

U[k]++;

}

R = new int[m];

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

R[i] = 0;

for (i = 0; i < m; i++)

{

j = D[i];

R[j]++;

}

P = new int[m];

k = -1; t = 0;

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

if (R[i] == 0)

{

k++; P[k] = i;

}

while (t < k)

{

i = P[t];

for (j = S[i]; j <= S[i] + L[i] - 1; j++)

{

q = D[j]; R[q]--;

if (R[q] == 0)

{

k++; P[k] = q;

}

}

t++;

}

if (k < n - 1)

cout << "Граф не ацикличен";

else

for (i = 0; i < k; i++)

{

cout << P[i] << endl;

}

}

Трудоемкость данного алгоритма для графа из n вершин и m ребер: O(n + m)