Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**Иркутский национальный исследовательский технический университет**

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |
|  |

|  |
| --- |
| **Отчет** |
| по лабораторной работе №4 по дисциплине «Дискретная математика»  « Программная реализация алгоритмов на графах.» |
| наименование темы |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил |  | АСУб-20-2 |  |  |  | Арбакова А.В. |
|  |  | шифр |  | подпись |  | И.О. Фамилия |
| Проверил |  |  |  |  |  | Л.Л. Носырева |
|  |  |  |  | подпись |  | И.О. Фамилия |

**Программная реализация алгоритмов на графах**

**Алгоритм Дейкстры**

**Цель работы:** Изучить способы моделирования алгоритмов на графах и научиться разрабатывать программы, реализующие алгоритмы на графах.

**Условие задачи:** Разработать компьютерную программу реализующую алгоритм согласно варианта.

**Листинг программы:**

#include <iostream>

using namespace std;

int main()

{

int sizeOfMatrix, min, indexMin, indexPeak;

string letter = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

cout << "Введите количество вершин: ";

cin >> sizeOfMatrix;

int pos[sizeOfMatrix], node[sizeOfMatrix];

int\*\* A;

A = new int\* [sizeOfMatrix];

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

A[i] = new int[sizeOfMatrix];

cout << "Введите длины рёбер:\n";

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

for (int j = 1 + i; j < sizeOfMatrix; j++)

{

cout << '[' << letter[i] << "][" << letter[j] << "] = ";

cin >> A[i][j];

A[j][i] = A[i][j];

}

cout << "Матрица смежности:\n";

cout << ' ';

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

{

cout.width(7);

cout << letter[i];

}

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

{

cout << endl;

cout << letter[i];

for (int j = 0; j < sizeOfMatrix; j++)

{

cout.width(3);

cout << A[i][j];

}

}

cout << endl;

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

{

pos[i] = 100000;

node[i] = 0;

}

cout << "Введите произвольную вершину: ";

char peak;

cin >> peak;

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

if (letter[i] == peak)

indexPeak = i;

pos[indexPeak] = 0;

for (int i = 0; i < sizeOfMatrix - 1; i++)

{

min = 100000;

for (int j = 0; j < sizeOfMatrix; j++)

{

if (!node[j] && pos[j] < min)

{

min = pos[j];

indexMin = j;

}

}

node[indexMin] = 1;

for (int j = 0; j < sizeOfMatrix; j++)

{

if (!node[j] && A[indexMin][j] > 0 && pos[indexMin] != 100000 && pos[indexMin] + A[indexMin][j] < pos[j])

{

pos[j] = pos[indexMin] + A[indexMin][j];

}

}

}

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

if (letter[indexPeak] != letter[i])

cout << "Кратчайший путь из вершины " << letter[indexPeak] <<" в вершину " << letter[i] << ": " << pos[i] << endl;

return 0;

}

**Тесты программы:**

|  |  |  |
| --- | --- | --- |
| № теста | Входные данные | Результат |
| 1 | Количество вершин:4 |  |
| 2 | Количество вершин:3 |  |