Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Институт информационных технологий

Факультет компьютерных технологий

Кафедра информационных систем и технологий

Лабораторная работа №13

по дисциплине

«Алгоритмы и структуры данных»

|  |  |
| --- | --- |
|  | Выполнил студент группы 181073  Наркевич Михаил Викторович |
|  | Проверил ассистент кафедры ИСиТ Потоцкий Дмитрий Сергеевич |

Минск 2022

Лабораторная работа № 13

Вариант 1

*Задание.* Представить ориентированный граф, состоящий из 7-10 вершин, с помощью матрицы смежности, а затем выполнить следующие операторы над его элементами. Указать вершину-источник, а затем решить следующие задачи.

1. Кратчайшие пути от вершины-источника до всех вершин орграфа на основе алгоритма Дейкстры.

Реализация программы (основной код):

Console.WriteLine("Задание 1. Кратчайшие пути от вершины-источника до всех вершин орграфа на основе алгоритма Дейкстры.");

Console.Write("Введите количество узлов графа: ");

var size = int.Parse(Console.ReadLine()!);

var matrix = new int[size, size];

for (var i = 0; i < size; i++)

{

for (var j = 0; j < size; j++)

{

Console.Write($"Введите значение дуги {i}->{j} или 0: ");

matrix[i, j] = int.Parse(Console.ReadLine()!);

}

}

Console.Write("Введите вершину-источник: ");

var source = int.Parse(Console.ReadLine()!);

Algorithm(matrix, source, size);

Console.ReadKey();

int MinDistance(IReadOnlyList<int> dist, IReadOnlyList<bool> sptSet)

{

var maxValue = int.MaxValue;

var minIndex = -1;

for (var i = 0; i < dist.Count; i++)

{

if (sptSet[i] || dist[i] > maxValue)

continue;

maxValue = dist[i];

minIndex = i;

}

return minIndex;

}

void Algorithm(int[,] matrix, int root, int size)

{

var dist = new int[size];

var path = new int[size];

var checkPoint = new bool[size];

for (var i = 0; i < size; i++)

{

dist[i] = int.MaxValue;

checkPoint[i] = false;

}

dist[root] = 0;

for (var i = 1; i < size; i++)

{

var maxValue = int.MaxValue;

var minDist = -1;

for (var j = 0; j < dist.Length; j++)

{

if (checkPoint[j] || dist[j] > maxValue)

continue;

maxValue = dist[j];

minDist = j;

}

checkPoint[minDist] = true;

for (var j = 0; j < size; j++)

{

if (checkPoint[j]

|| matrix[minDist, j] == 0

|| dist[minDist] == int.MaxValue

|| dist[minDist] + matrix[minDist, j] >= dist[j])

continue;

dist[j] = dist[minDist] + matrix[minDist, j];

path[j] = minDist;

}

if (path[i] == 0)

Console.WriteLine($"Кратчайший путь: из {root}->{i} | Расстояние: {dist[i]}");

else

{

var stack = new Stack<int>();

stack.Push(path[i]);

Console.Write($"Кратчайший путь: из {root}->");

for (var j = path[i]; j != 0; j = path[j])

{

if (path[j] == 0)

break;

stack.Push(path[j]);

j = path[j];

}

for (var j = 0; j <= stack.Count; j++)

{

if (j == stack.Count)

Console.Write($"{i} | Расстояние: {dist[i]}");

else

{

Console.Write(stack.Pop() + " -> ");

j = -1;

}

}

Console.WriteLine();

}

}

}

Программный вывод:

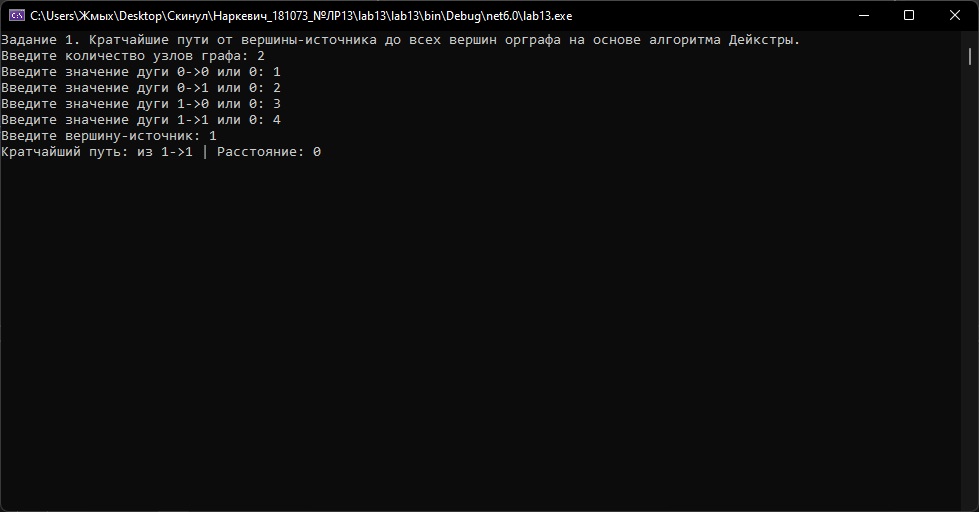


Рисунок 1 – Алгоритм Дейкстры

Контрольные вопросы и задания

1.       Для решения какой задачи на ориентированном графе удобно использовать алгоритм Дейкстры?

Находит кратчайшие пути от одной из вершин графа до всех остальных. Алгоритм работает только для графов без рёбер отрицательного веса. Алгоритм широко применяется в программировании и технологиях данного мира, например, его используют протоколы маршрутизации OSPF и IS-IS.

2.       Для решения какой задачи на ориентированном графе удобно использовать алгоритм Флойда?

Алгоритм Флойда может быть использован для решения следующих задач, в частности:

* кратчайшие пути в ориентированных графах;
* [транзитивное замыкание](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D0%B7%D0%B8%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5_%D0%B7%D0%B0%D0%BC%D1%8B%D0%BA%D0%B0%D0%BD%D0%B8%D0%B5) ориентированных графов;
* нахождение [регулярного выражения](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D1%83%D0%BB%D1%8F%D1%80%D0%BD%D1%8B%D0%B5_%D0%B2%D1%8B%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F), обозначающего [регулярный язык](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D1%83%D0%BB%D1%8F%D1%80%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA), принимаемый [конечным автоматом](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%B5%D1%87%D0%BD%D1%8B%D0%B9_%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82);
* [обращение](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B2%D1%8B%D1%80%D0%BE%D0%B6%D0%B4%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BC%D0%B0%D1%82%D1%80%D0%B8%D1%86%D0%B0) [вещественных](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D1%89%D0%B5%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE) [матриц](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D1%80%D0%B8%D1%86%D0%B0_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0));
* оптимальная маршрутизация. В этом приложении нужно найти путь с максимальным потоком между двумя вершинами. Это означает, что вместо взятия минимумов, как в псевдокоде выше, используются максимумы. Веса ребер представляют фиксированные ограничения для потока. Веса путей представляют собой узкие места поэтому операция сложения, указанная выше, заменяется минимальной операцией;
* быстрый расчет [сетей Pathfinder](https://en.wikipedia.org/wiki/Pathfinder_network);
* [самые широкие пути/пути с максимальной пропускной способностью](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BE_%D1%81%D0%B0%D0%BC%D0%BE%D0%BC_%D1%88%D0%B8%D1%80%D0%BE%D0%BA%D0%BE%D0%BC_%D0%BF%D1%83%D1%82%D0%B8);
* вычисление канонической формы матриц разностных границ;
* вычисление сходства между графами.

3.       Какая вершина графа называется его центром?

Центром орграфа G называется вершина с минимальным эксцентриситетом, т.е. это вершина, для которой максимальное расстояние (длина пути) до других вершин минимально.