

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ  
БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

ФАКУЛЬТЕТ ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Кафедра вычислительных систем

## **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

Выполнил:  
студент группы ИВ-521  
Прокопенко Р.П.

Проверил:  
доцент кафедры вычислительных систем  
Куринов М. Г.

Оценка – «\_\_\_\_\_».

Новосибирск - 2016

### Постановка задачи

Требуется реализовать алгоритм Дейкстры поиска кратчайшего пути от заданной вершины до всех остальных. Для хранения длин кратчайших путей (массив  $d[i]$ ) следует использовать бинарную кучу (min-heap). Для хранения графа использовать матрицу смежности.

### Описание алгоритмов

Алгоритм Дейкстры (англ. *Dijkstra's algorithm*) — алгоритм на графах, изобретённый нидерландским учёным Эдсгером Дейкстрой в 1959 году. Находит кратчайшие пути от одной из вершин графа до всех остальных. Алгоритм работает только для графов без рёбер отрицательного веса.

В начале алгоритма расстояние для начальной вершины полагается равным нулю, а все остальные расстояния заполняются большим положительным числом (большим максимального возможного пути в графе). Массив флагов заполняется нулями. Затем запускается основной цикл.

На каждом шаге цикла мы ищем вершину  $u$  с минимальным расстоянием и флагом равным нулю. Затем мы устанавливаем в ней флаг в 1 и проверяем все соседние с ней вершины  $v$ . Если в них (в  $v$ ) расстояние больше, чем сумма расстояния до текущей вершины и длины ребра, то уменьшаем его. Цикл завершается, когда флаги всех вершин становятся равны 1, либо когда у всех вершин с флагом 0.  $d[i] = \infty$  последний случай возможен тогда и только тогда, когда граф  $G$  несвязный.

Оценка сложности:

$$T_{\text{дейкстры}} = O(n \log n + m \log n)$$

$Dd[i]$  хранятся в бинарной куче (Binary heap)

$$T_{\text{дейкстра}} = O(m \log n)$$

#### **Binary heap**

$$T_{\text{create}} = O(1)$$

$$T_{\text{min}} = O(1)$$

$$T_{\text{insert}} = O(\log n)$$

$$T_{\text{heapify}} = O(\log n)$$

$$T_{\text{decrease}} = O(\log n)$$

## Экспериментальное исследование

Таблица1 - Таблица инцидентности

-	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	∞	3	∞	∞	∞	∞	10	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	3	∞	2
2	3	∞	∞	∞	∞	∞	∞	∞	4	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	8	∞	∞
4	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	7	∞	∞	∞	∞	∞
5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	8	7	∞	∞	∞	∞	∞	∞	∞	∞
6	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	5	∞	∞	∞
7	10	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	3	∞	∞	2	∞	∞	∞
8	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	9	∞	∞	∞	∞	∞
9	∞	4	∞	∞	∞	∞	∞	∞	∞	2	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
10	∞	∞	∞	∞	∞	∞	∞	∞	2	∞	∞	∞	∞	11	5	∞	∞	∞	∞	∞
11	∞	∞	∞	∞	8	∞	∞	∞	∞	∞	∞	∞	4	∞	∞	∞	∞	∞	∞	∞
12	∞	∞	∞	∞	7	∞	∞	∞	∞	∞	∞	∞	7	∞	∞	∞	∞	∞	∞	1
13	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	4	7	∞	∞	∞	∞	∞	1	∞	∞
14	∞	∞	∞	∞	∞	∞	3	∞	∞	11	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
15	∞	∞	∞	7	∞	∞	∞	9	∞	5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
16	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	5	∞
17	∞	∞	∞	∞	∞	5	2	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	6	∞
18	3	∞	8	∞	∞	∞	∞	∞	∞	∞	∞	∞	1	∞	∞	∞	∞	∞	∞	∞
19	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	5	6	∞	∞	4
20	2	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1	∞	∞	∞	∞	∞	∞	4	∞

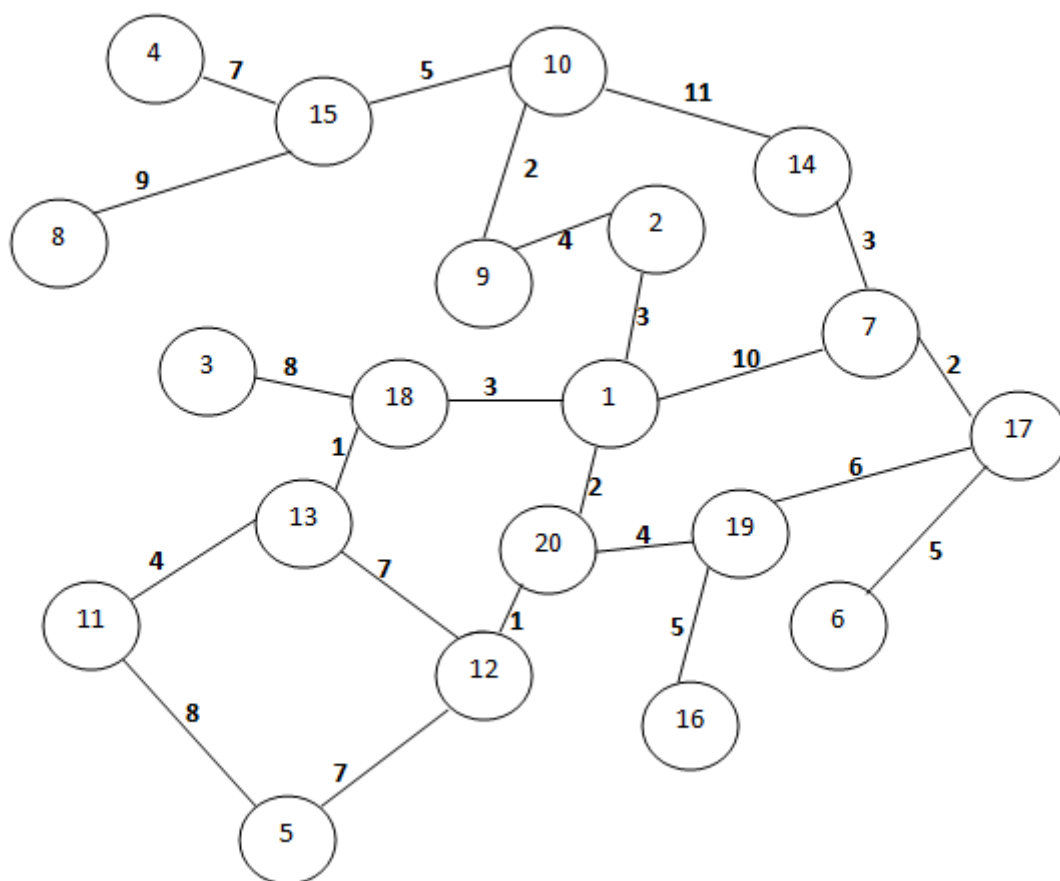


Рисунок 1 - Граф

После выполнения программы были получены следующие результаты:

Min path from 1 to 2 partlen: 3 path 1 2	Min path from 1 to 8 partlen: 23 path 1 2 9 10 15 8	Min path from 1 to 14 partlen: 13 path 1 7 14
Min path from 1 to 3 partlen: 11 path 1 18 3	Min path from 1 to 9 partlen: 7 path 1 2 9	Min path from 1 to 15 partlen: 14 path 1 2 9 10 15
Min path from 1 to 4 partlen: 21 path 1 2 9 10 15 4	Min path from 1 to 10 partlen: 9 path 1 2 9 10	Min path from 1 to 16 partlen: 11 path 1 20 19 16
Min path from 1 to 5 partlen: 10 path 1 20 12 5	Min path from 1 to 11 partlen: 8 path 1 18 13 11	Min path from 1 to 17 partlen: 12 path 1 20 19 17
Min path from 1 to 6 partlen: 17 path 1 20 19 17 6	Min path from 1 to 12 partlen: 3 path 1 20 12	Min path from 1 to 18 partlen: 3 path 1 18
Min path from 1 to 7 partlen: 10 path 1 7	Min path from 1 to 13 partlen: 4 path 1 18 13	Min path from 1 to 19 partlen: 6 path 1 20 19
Min path from 1 to 20 partlen: 2 path 1 20		

### **Выводы**

Были реализованы алгоритмы для работы с бинарной кучей, алгоритм Дейкстры, алгоритмы работы с графами (создание графа, добавление ребра и др.). С помощью алгоритма Дейкстра были рассчитаны минимальные пути и вершины 1 во все остальные 19 вершин (2-19).

### **Ссылки**

1. <http://www.mkurnosov.net/teaching/>
2. [https://ru.wikipedia.org/wiki/Алгоритм\\_Дейкстры](https://ru.wikipedia.org/wiki/Алгоритм_Дейкстры)