Отчет о выполнении задания 3  
«Алгоритм Дейкстры»

**Мингазеев Роман Лутфиевич**

Группа ИВ-521

direnol@yandex.ru

**Описание алгоритмов**

**Бинарное дерево поиска**

**Алгори́тм Де́йкстры**(англ.*Dijkstra’s algorithm*) — алгоритм на графах, изобретённый нидерландским учёны 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 (бо́льшим 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 *u*. Если в них (в *u*) 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 c флагом 0. *d*[*i*] = *∞* последний случай возможен тогда и только тогда, когда граф *G* несвязный.

**Оценка сложности** :

Tдейкстры = *O*(*n* log *n* + *m* log *n*)

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

Tдейкстра = *O*(*m* log *n*)

**Binary heap**

*T*create = *O*(1)

*T*min = *O*(1)

*Ti*nsert =*O*(log *n*)

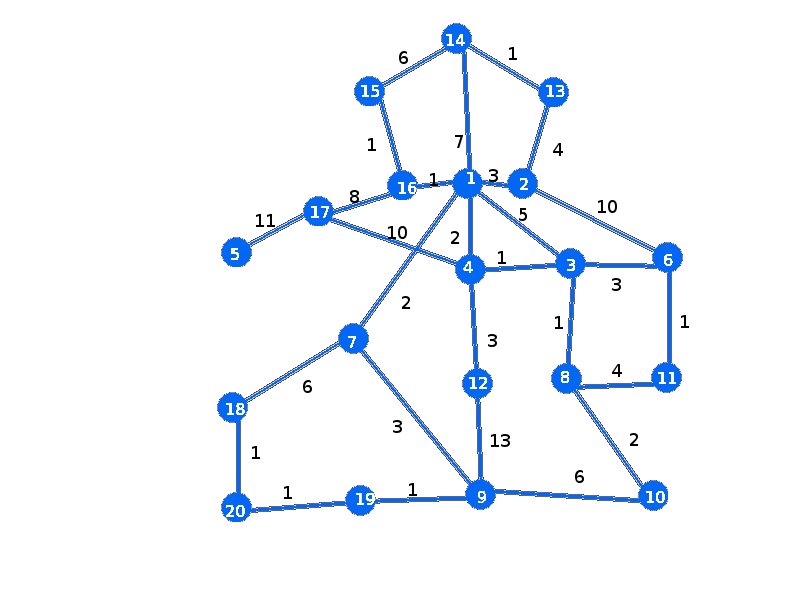
*T*heapify = *O*(log *n*)

*Tde*crease = *O*(log *n*

**Экспериментальное исследование**

**Таблица инцидентнсости.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **-** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **1** | **0** | **3** | **5** | **2** | **∞** | **∞** | **6** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **7** | **∞** | **1** | **∞** | **∞** | **∞** | **∞** |
| **2** | **3** | **0** | **∞** | **∞** | **∞** | **10** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **4** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** |
| **3** | **5** | **∞** | **0** | **1** | **∞** | **3** | **∞** | **1** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** |
| **4** | **2** | **∞** | **1** | **0** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **3** | **∞** | **∞** | **∞** | **∞** | **10** | **∞** | **∞** | **∞** |
| **5** | **∞** | **∞** | **∞** | **∞** | **0** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **11** | **∞** | **∞** | **∞** |
| **6** | **∞** | **10** | **3** | **∞** | **∞** | **0** | **∞** | **∞** | **∞** | **∞** | **1** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** |
| **7** | **6** | **∞** | **∞** | **∞** | **∞** | **∞** | **0** | **∞** | **3** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **6** | **∞** | **∞** |
| **8** | **∞** | **∞** | **1** | **∞** | **∞** | **∞** | **∞** | **0** | **∞** | **2** | **4** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** |
| **9** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **3** | **∞** | **0** | **6** | **∞** | **13** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **1** | **∞** |
| **10** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **2** | **6** | **0** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** |
| **11** | **∞** | **∞** | **∞** | **∞** | **∞** | **1** | **∞** | **4** | **∞** | **∞** | **0** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** |
| **12** | **∞** | **∞** | **∞** | **3** | **∞** | **∞** | **∞** | **∞** | **13** | **∞** | **∞** | **0** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** |
| **13** | **∞** | **4** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **0** | **1** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** |
| **14** | **7** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **1** | **0** | **6** | **∞** | **∞** | **∞** | **∞** | **∞** |
| **15** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **6** | **0** | **1** | **∞** | **∞** | **∞** | **∞** |
| **16** | **1** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **1** | **0** | **8** | **∞** | **∞** | **∞** |
| **17** | **∞** | **∞** | **∞** | **10** | **11** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **8** | **0** | **∞** | **∞** | **∞** |
| **18** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **0** | **∞** | **1** |
| **19** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **1** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **0** | **1** |
| **20** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **∞** | **1** | **1** | **0** |

**Граф.**

**Min path from** 1 to 2

partlen: 3

path 1 2

**Min path from** 1 to 3

partlen: 3

path 1 4 3

**Min path from** 1 to 4

partlen: 2

path 1 4

**Min path from** 1 to 5

partlen: 20

path 1 16 17 5

**Min path from** 1 to 6

partlen: 6

path 1 4 3 6

**Min path from** 1 to 7

partlen: 6

path 1 7

**Min path from** 1 to 8

partlen: 4

path 1 4 3 8

**Min path from** 1 to 9

partlen: 9

path 1 7 9

**Min path from** 1 to 10

partlen: 6

path 1 4 3 8 10

**Min path from** 1 to 11

partlen: 7

path 1 4 3 6 11

**Min path from** 1 to 12

partlen: 5

path 1 4 12

**Min path from** 1 to 13

partlen: 7

path 1 2 13

**Min path from** 1 to 14

partlen: 7

path 1 14

**Min path from** 1 to 15

partlen: 2

path 1 16 15

**Min path from** 1 to 16

partlen: 1

path 1 16

**Min path from** 1 to 17

partlen: 9

path 1 16 17

**Min path from** 1 to 18

partlen: 12

path 1 7 18

**Min path from** 1 to 19

partlen: 10

path 1 7 9 19

**Min path from** 1 to 20

partlen: 11

path 1 7 9 19 20

**Вывод**

Реализовав алгоритм дейкстра с использованием Binary Heap, мы нашли крайтчайшие пути из 1 вершины до остальных.

**Ссылки**

1. <http://www.mkurnosov.net/teaching/>