

Module `dijkstra`

► [EXPAND SOURCE CODE](#)

Functions

```
def parser(V, E)
```

Глобальная функция, производящая множество объектов класса "Ребро" из множеств вершин и ребёр, подаваемых на вход и оформленных в стиле условия из учебника.

► [EXPAND SOURCE CODE](#)

```
def png_to_gif(V)
```

Глобальная функция, производящая конвертацию множества картинок формата png в один файл формата gif с задержкой между кадрами в 1 секунду

► [EXPAND SOURCE CODE](#)

```
def run(N)
```

Глобальная функция запуска модуля

► [EXPAND SOURCE CODE](#)

Classes

```
class Graph (n, bias, i_start, i_finish)
```

Класс Граф

► [EXPAND SOURCE CODE](#)

Instance variables

```
var bias
```

Значение N из условия задачи.

```
var i_finish
```

Индекс вершины, в которую нужно найти кратчайший путь.

```
var i_start
```

Индекс вершины, из которой нужно найти кратчайший путь.

```
var n
```

Количество вершин в графе.

```
var paths
```

Массив (при инициализации содержащий только пометку стартовой вершины), который в начале поиска будет заполнен нулями. В процессе поиска кратчайшего пути в ячейке вершины, входящей в кратчайший

путь, записывается номер ячейки связанной вершины.

```
var unseen
```

Массив вершин, не пройденных в процессе работы алгоритма.

```
var vertexes
```

Массив объектов класса "Вершина" в графе.

Methods

```
def generate_frame(self, iter, current)
```

Метод, производящий рисование состояния графа при помощи модуля GraphViz и конвертации *.dot-файлов* в *.png-файлы* утилитой *dot.exe*. На вход подаются номер итерации (0 - состояние графа до работы алгоритма) и индекс вершины, рассматриваемой на данной итерации.

► [EXPAND SOURCE CODE](#)

```
def generate_last_frame(self, iter, path, final_weight)
```

Метод, производящий рисование состояния графа после работы алгоритма, аналогично `generate_frame`, но с выводом общего веса кратчайшего пути сверху, а также закрашиванием вершин, входящих в кратчайший путь.

► [EXPAND SOURCE CODE](#)

```
def lookup_pair(self, v, i)
```

Метод, проверяющий, есть ли у конкретного объекта класса "Вершина", подаваемого на вход, связь с вершиной, индекс которой подаётся на вход, а также возвращающий `True`, если связь есть, и `False`, если связи нет.

► [EXPAND SOURCE CODE](#)

```
def min_w(self)
```

Метод, выполняющий поиск ближайшей вершины и веса пути до неё. Возвращает индекс вершины и вес пути до неё.

► [EXPAND SOURCE CODE](#)

```
def route(self)
```

Метод, выстраивающий кратчайший путь из полученных расчётов и заполненного массива `paths[]` и возвращающий массив индексов вершин кратчайшего пути в порядке от стартовой до конечной вершин.

► [EXPAND SOURCE CODE](#)

```
def search(self)
```

Метод, выполняющий расчёт общего веса кратчайшего пути, который возвращается на выходе.

► [EXPAND SOURCE CODE](#)

```
def update_links(self, links)
```

Метод, распределяющий данные о вершинах и их связях из массива объектов класса "Рёбро", подаваемого на вход, а также рассчитывающий веса рёбер.

► [EXPAND SOURCE CODE](#)

```
class Link (i, j, bdir)
```

Класс Ребро

► [EXPAND SOURCE CODE](#)

Instance variables

```
var bdir
```

Булево значение направленности ребра (однонаправленное - False, двунаправленное - True).

```
var i
```

Индекс первой вершины в ребре.

```
var j
```

Индекс второй вершины в ребре.

```
class Vertex
```

Класс Вершина

► [EXPAND SOURCE CODE](#)

Instance variables

```
var done
```

Данное значение показывает, рассмотрена ли вершина в процессе работы алгоритма, для того, чтобы избежать бесконечного выполнения алгоритма.

```
var olinks
```

Двумерный массив: множество массивов длиной 2, включающие в себя вершины, с которыми данная вершина связана рёбрами, а также веса ребёр до этих вершин, включены в единый массив.

```
var w
```

Вес пути до вершины от старта. Принимает значение maxsize до тех пор, пока не пройдёт итерация в алгоритме по этой вершине.

Dijkstra Algorithm

Perfomed by Pavel 6az00ka ([Telegram](#)).

Check the timestamps of commits in my public repository:

[\[Open GitHub\]](#)

Functions

[parser](#)

[png_to_gif](#)

[run](#)

Classes

Graph

`bias`
`generate_frame`
`generate_last_frame`
`i_finish`
`i_start`
`lookup_pair`
`min_w`
`n`
`paths`
`route`
`search`
`unseen`
`update_links`
`vertexes`

Link

`bdir`
`i`
`j`

Vertex

`done`
`olinks`
`w`