

## GraphAssignment1

Generated by Doxygen 1.9.3



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy . . . . .	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List . . . . .	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 repo.EdgeIterator Class Reference . . . . .	5
3.1.1 Detailed Description . . . . .	5
3.2 repo.Graph Class Reference . . . . .	6
3.2.1 Detailed Description . . . . .	6
3.2.2 Member Function Documentation . . . . .	6
3.2.2.1 add_cost() . . . . .	6
3.2.2.2 add_edge() . . . . .	7
3.2.2.3 add_vertex() . . . . .	7
3.2.2.4 check_cost() . . . . .	7
3.2.2.5 check_edge() . . . . .	7
3.2.2.6 check_vertex() . . . . .	8
3.2.2.7 count_edges() . . . . .	8
3.2.2.8 count_vertices() . . . . .	8
3.2.2.9 delete_cost() . . . . .	8
3.2.2.10 delete_edge() . . . . .	9
3.2.2.11 delete_vertex() . . . . .	9
3.2.2.12 file_read_new() . . . . .	9
3.2.2.13 file_read_original() . . . . .	9
3.2.2.14 file_write() . . . . .	10
3.2.2.15 list_in_vert() . . . . .	10
3.2.2.16 list_isolated_vert() . . . . .	10
3.2.2.17 list_out_vert() . . . . .	10
3.3 repo.RepositoryException Class Reference . . . . .	11
3.4 ui.UI Class Reference . . . . .	11
3.4.1 Detailed Description . . . . .	11
3.4.2 Member Function Documentation . . . . .	11
3.4.2.1 add_edge() . . . . .	12
3.4.2.2 add_vertex() . . . . .	12
3.4.2.3 change_cost() . . . . .	12
3.4.2.4 check_edge() . . . . .	12
3.4.2.5 check_vertex() . . . . .	12
3.4.2.6 console() . . . . .	13
3.4.2.7 count_edges() . . . . .	13
3.4.2.8 count_vertices() . . . . .	13
3.4.2.9 create_random() . . . . .	13
3.4.2.10 del_edge() . . . . .	13

3.4.2.11 del_vertex()	14
3.4.2.12 file_io()	14
3.4.2.13 list_degree()	14
3.4.2.14 list_inbound()	14
3.4.2.15 list_inbound_nr()	14
3.4.2.16 list_outbound()	15
3.4.2.17 list_outbound_nr()	15
3.4.2.18 print_graph()	15
3.4.2.19 read100k()	15
3.4.2.20 read100k_modified()	16
3.4.2.21 read10k()	16
3.4.2.22 read10k_modified()	16
3.4.2.23 read1k()	16
3.4.2.24 read1k_modified()	16
3.4.2.25 write100k()	17
3.4.2.26 write10k()	17
3.4.2.27 write1k()	17

<b>Index</b>	<b>19</b>
--------------	-----------

# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

repo.EdgeIterator . . . . .	5
Exception	
repo.RepositoryException . . . . .	11
repo.Graph . . . . .	6
ui.UI . . . . .	11



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<b>repo.EdgeIterator</b>	5
<b>repo.Graph</b>	6
<b>repo.RepositoryException</b>	11
<b>ui.UI</b>	11





## Chapter 3

# Class Documentation

### 3.1 repo.EdgeIterator Class Reference

#### Public Member Functions

- `def __init__(self, vertex_list)`
- `def __iter__(self)`
- `def __next__(self)`

#### Public Attributes

- `list`
- `value`
- `end`

#### 3.1.1 Detailed Description

This class is an iterator, iterating over a list representing inbound or outbound edges

The documentation for this class was generated from the following file:

- `GraphsLab1/repo.py`

## 3.2 repo.Graph Class Reference

### Public Member Functions

- `def __init__ (self)`
- `def file_read_original (self, file)`
- `def file_read_new (self, file)`
- `def file_write (self, file)`
- `def count_vertices (self)`
- `def count_edges (self)`
- `def list_isolated_vert (self)`
- `def list_in_vert (self, vertex)`
- `def list_out_vert (self, vertex)`
- `def add_vertex (self, vertex)`
- `def check_vertex (self, vertex)`
- `def delete_vertex (self, vertex)`
- `def check_cost (self, vertex1, vertex2)`
- `def add_cost (self, vertex1, vertex2, cost)`
- `def delete_cost (self, vertex1, vertex2)`
- `def check_edge (self, vertex1, vertex2)`
- `def add_edge (self, vertex1, vertex2)`
- `def delete_edge (self, vertex1, vertex2)`

### Public Attributes

- `Din`
- `Dout`
- `Dcost`

### 3.2.1 Detailed Description

Implementation of the Graph.

Uses three dictionaries for storing values.

Din and Dout have all existing vertices as keys, and a corresponding list of inbound / outbound vertices as values

Dcost has a pairs of two vertices as keys, and integers as values

### 3.2.2 Member Function Documentation

#### 3.2.2.1 add\_cost()

```
def repo.Graph.add_cost (  
    self,  
    vertex1,  
    vertex2,  
    cost )
```

This function add a cost to an edge

Checks if edge exists

:param vertex1: integer

:param vertex2: integer

:param cost: integer

:return: RepositoryException if edge doesn't exist

### 3.2.2.2 add\_edge()

```
def repo.Graph.add_edge (  
    self,  
    vertex1,  
    vertex2 )
```

This function adds an edge to the graph  
Checks if edge already exists  
:param vertex1: integer  
:param vertex2: integer  
:return: nothing

### 3.2.2.3 add\_vertex()

```
def repo.Graph.add_vertex (  
    self,  
    vertex )
```

Adds a vertex to the Graph  
Checks if vertex already exists  
:param vertex: integer  
:return: RepositoryException if vertex exists

### 3.2.2.4 check\_cost()

```
def repo.Graph.check_cost (  
    self,  
    vertex1,  
    vertex2 )
```

This function return the cost of an edge  
:param vertex1: integer  
:param vertex2: integer  
:return: integer

### 3.2.2.5 check\_edge()

```
def repo.Graph.check_edge (  
    self,  
    vertex1,  
    vertex2 )
```

This function checks if an edge exists  
:param vertex1: integer  
:param vertex2: integer  
:return: boolean

### 3.2.2.6 check\_vertex()

```
def repo.Graph.check_vertex (
    self,
    vertex )
```

Checks if vertex already exists  
:param vertex: integer  
:return: RepositoryException if vertex exists

### 3.2.2.7 count\_edges()

```
def repo.Graph.count_edges (
    self )
```

Returns number of edges in matrix  
:return: integer

### 3.2.2.8 count\_vertices()

```
def repo.Graph.count_vertices (
    self )
```

Simply returns the number of vertices in the matrix  
:return: integer

### 3.2.2.9 delete\_cost()

```
def repo.Graph.delete_cost (
    self,
    vertex1,
    vertex2 )
```

This function deletes a cost from an edge  
Checks if edge exists  
:param vertex1: integer  
:param vertex2: integer  
:return: RepositoryException if edge doesn't exist

### 3.2.2.10 delete\_edge()

```
def repo.Graph.delete_edge (
    self,
    vertex1,
    vertex2 )
```

This function deletes an edge from the graph  
Checks if edge exists  
:param vertex1: integer  
:param vertex2: integer  
:return: nothing

### 3.2.2.11 delete\_vertex()

```
def repo.Graph.delete_vertex (
    self,
    vertex )
```

Deletes a vertex from the Graph  
Checks if vertex doesn't exist  
Also deletes all edges connected to said vertex  
:param vertex: integer  
:return: RepositoryException if vertex doesn't exist

### 3.2.2.12 file\_read\_new()

```
def repo.Graph.file_read_new (
    self,
    file )
```

This function reads a graph from files written by the program  
:param file: the file to read from  
:return: nothing

### 3.2.2.13 file\_read\_original()

```
def repo.Graph.file_read_original (
    self,
    file )
```

This function reads files in the original format  
:param file: the file to read from  
:return: nothing

#### 3.2.2.14 file\_write()

```
def repo.Graph.file_write (
    self,
    file )
```

Writes Graph to file like this:

Three values on each row, the first two represent the ends of an edge, the third is the cost  
Then if there are any remaining vertices with no edges, they are listed at the end of the file  
:param file: the file to write to  
:return: nothing

#### 3.2.2.15 list\_in\_vert()

```
def repo.Graph.list_in_vert (
    self,
    vertex )
```

Returns list of inbound edges of a vertex  
:param vertex: integer  
:return: a list of integers

#### 3.2.2.16 list\_isolated\_vert()

```
def repo.Graph.list_isolated_vert (
    self )
```

Returns a list of vertices with no inbound or outbound edges  
:return: a list of integers

#### 3.2.2.17 list\_out\_vert()

```
def repo.Graph.list_out_vert (
    self,
    vertex )
```

Returns list of outbound edges of a vertex  
:param vertex: integer  
:return: a list of integers

The documentation for this class was generated from the following file:

- GraphsLab1/repo.py

## 3.3 repo.RepositoryException Class Reference

Inherits Exception.

The documentation for this class was generated from the following file:

- GraphsLab1/repo.py

## 3.4 ui.UI Class Reference

### Public Member Functions

- def **\_\_init\_\_** (self, repo)
- def **console** (self)
- def **count\_vertices** (self)
- def **count\_edges** (self)
- def **add\_vertex** (self)
- def **check\_vertex** (self)
- def **del\_vertex** (self)
- def **add\_edge** (self)
- def **check\_edge** (self)
- def **del\_edge** (self)
- def **change\_cost** (self)
- def **list\_inbound** (self)
- def **list\_inbound\_nr** (self, vertex)
- def **list\_outbound** (self)
- def **list\_outbound\_nr** (self, vertex)
- def **list\_degree** (self)
- def **print\_graph** (self)
- def **file\_io** (self)
- def **read1k** (self)
- def **read10k** (self)
- def **read100k** (self)
- def **read1k\_modified** (self)
- def **read10k\_modified** (self)
- def **read100k\_modified** (self)
- def **write1k** (self)
- def **write10k** (self)
- def **write100k** (self)
- def **create\_random** (self)

### 3.4.1 Detailed Description

This is the Main UI function

### 3.4.2 Member Function Documentation

#### 3.4.2.1 add\_edge()

```
def ui.UI.add_edge (
    self )
```

This function adds an edge using add\_edge repo function  
:return: nothing

#### 3.4.2.2 add\_vertex()

```
def ui.UI.add_vertex (
    self )
```

Adds a vertex using add\_vertex from repo  
:return: nothing

#### 3.4.2.3 change\_cost()

```
def ui.UI.change_cost (
    self )
```

This function modifies the cost of an edge  
:return: nothing

#### 3.4.2.4 check\_edge()

```
def ui.UI.check_edge (
    self )
```

This function checks existence of an edge  
:return: nothing, result is printed in console

#### 3.4.2.5 check\_vertex()

```
def ui.UI.check_vertex (
    self )
```

This function checks existence of a vertex  
:return: nothing, and result is printed in console



#### 3.4.2.6 console()

```
def ui.UI.console (
    self )
```

This function lets the user choose one of the listed options  
:return: nothing

#### 3.4.2.7 count\_edges()

```
def ui.UI.count_edges (
    self )
```

This function returns the number of edges  
:return: nothing

#### 3.4.2.8 count\_vertices()

```
def ui.UI.count_vertices (
    self )
```

This function prints the number of vertices  
:return: nothing

#### 3.4.2.9 create\_random()

```
def ui.UI.create_random (
    self )
```

This function generates a random Graph after reading number of vertices and Edges from the console  
:return:

#### 3.4.2.10 del\_edge()

```
def ui.UI.del_edge (
    self )
```

This function deletes an edge  
:return: nothing

#### 3.4.2.11 del\_vertex()

```
def ui.UI.del_vertex (  
    self )
```

This function deletes a vertex  
:return: nothing

#### 3.4.2.12 file\_io()

```
def ui.UI.file_io (  
    self )
```

This function lets the user choose IO operations  
:return: nothing

#### 3.4.2.13 list\_degree()

```
def ui.UI.list_degree (  
    self )
```

This function prints the inbound and outbound degrees of a vertex  
:return: nothing

#### 3.4.2.14 list\_inbound()

```
def ui.UI.list_inbound (  
    self )
```

This function prints inbound vertices of given vertex  
:return: nothing

#### 3.4.2.15 list\_inbound\_nr()

```
def ui.UI.list_inbound_nr (  
    self,  
    vertex )
```

This function prints the inbound degree of a vertex  
:param vertex: integer  
:return: nothing

**3.4.2.16 list\_outbound()**

```
def ui.UI.list_outbound (
    self )
```

This function prints outbound vertices of given vertex  
:return: nothing

**3.4.2.17 list\_outbound\_nr()**

```
def ui.UI.list_outbound_nr (
    self,
    vertex )
```

This function prints the outbound degree of a vertex  
:param vertex: integer  
:return: nothing

**3.4.2.18 print\_graph()**

```
def ui.UI.print_graph (
    self )
```

This function prints the whole graph. Do not use this.  
:return: nothing

**3.4.2.19 read100k()**

```
def ui.UI.read100k (
    self )
```

This function reads a graph from file graph100k.txt  
:return: nothing

#### 3.4.2.20 read100k\_modified()

```
def ui.UI.read100k_modified (
    self )
```

This function reads a graph from file graph100knew.txt  
:return: nothing

#### 3.4.2.21 read10k()

```
def ui.UI.read10k (
    self )
```

This function reads a graph from file graph10k.txt  
:return: nothing

#### 3.4.2.22 read10k\_modified()

```
def ui.UI.read10k_modified (
    self )
```

This function reads a graph from file graph10knew.txt  
:return: nothing

#### 3.4.2.23 read1k()

```
def ui.UI.read1k (
    self )
```

This function reads a graph from file graph1k.txt  
:return: nothing

#### 3.4.2.24 read1k\_modified()

```
def ui.UI.read1k_modified (
    self )
```

This function reads a graph from file graph1knew.txt  
:return: nothing

#### 3.4.2.25 write100k()

```
def ui.UI.write100k (
    self )
```

This function writes the graph to graph100knew.txt  
:return: nothing

#### 3.4.2.26 write10k()

```
def ui.UI.write10k (
    self )
```

This function writes the graph to graph10knew.txt  
:return: nothing

#### 3.4.2.27 write1k()

```
def ui.UI.write1k (
    self )
```

This function writes the graph to graph1knew.txt  
:return: nothing

The documentation for this class was generated from the following file:

- GraphsLab1/ui.py



# Index

- add\_cost
  - repo.Graph, 6
- add\_edge
  - repo.Graph, 6
  - ui.UI, 11
- add\_vertex
  - repo.Graph, 7
  - ui.UI, 12
- change\_cost
  - ui.UI, 12
- check\_cost
  - repo.Graph, 7
- check\_edge
  - repo.Graph, 7
  - ui.UI, 12
- check\_vertex
  - repo.Graph, 7
  - ui.UI, 12
- console
  - ui.UI, 12
- count\_edges
  - repo.Graph, 8
  - ui.UI, 13
- count\_vertices
  - repo.Graph, 8
  - ui.UI, 13
- create\_random
  - ui.UI, 13
- del\_edge
  - ui.UI, 13
- del\_vertex
  - ui.UI, 13
- delete\_cost
  - repo.Graph, 8
- delete\_edge
  - repo.Graph, 8
- delete\_vertex
  - repo.Graph, 9
- file\_io
  - ui.UI, 14
- file\_read\_new
  - repo.Graph, 9
- file\_read\_original
  - repo.Graph, 9
- file\_write
  - repo.Graph, 9
- list\_degree
  - ui.UI, 14
- list\_in\_vert
  - repo.Graph, 10
- list\_inbound
  - ui.UI, 14
- list\_inbound\_nr
  - ui.UI, 14
- list\_isolated\_vert
  - repo.Graph, 10
- list\_out\_vert
  - repo.Graph, 10
- list\_outbound
  - ui.UI, 14
- list\_outbound\_nr
  - ui.UI, 15
- print\_graph
  - ui.UI, 15
- read100k
  - ui.UI, 15
- read100k\_modified
  - ui.UI, 15
- read10k
  - ui.UI, 16
- read10k\_modified
  - ui.UI, 16
- read1k
  - ui.UI, 16
- read1k\_modified
  - ui.UI, 16
- repo.EdgeIterator, 5
- repo.Graph, 6
  - add\_cost, 6
  - add\_edge, 6
  - add\_vertex, 7
  - check\_cost, 7
  - check\_edge, 7
  - check\_vertex, 7
  - count\_edges, 8
  - count\_vertices, 8
  - delete\_cost, 8
  - delete\_edge, 8
  - delete\_vertex, 9
  - file\_read\_new, 9
  - file\_read\_original, 9
  - file\_write, 9
  - list\_in\_vert, 10
  - list\_isolated\_vert, 10
  - list\_out\_vert, 10

repo.RepositoryException, 11

ui.UI, 11

- add\_edge, 11
- add\_vertex, 12
- change\_cost, 12
- check\_edge, 12
- check\_vertex, 12
- console, 12
- count\_edges, 13
- count\_vertices, 13
- create\_random, 13
- del\_edge, 13
- del\_vertex, 13
- file\_io, 14
- list\_degree, 14
- list\_inbound, 14
- list\_inbound\_nr, 14
- list\_outbound, 14
- list\_outbound\_nr, 15
- print\_graph, 15
- read100k, 15
- read100k\_modified, 15
- read10k, 16
- read10k\_modified, 16
- read1k, 16
- read1k\_modified, 16
- write100k, 16
- write10k, 17
- write1k, 17

write100k

- ui.UI, 16

write10k

- ui.UI, 17

write1k

- ui.UI, 17