

Graphs

Generated by Doxygen 1.8.17

1 README	1
2 Todo List	3
3 Data Structure Index	5
3.1 Data Structures	5
4 File Index	7
4.1 File List	7
5 Data Structure Documentation	9
5.1 Graph Struct Reference	9
5.1.1 Detailed Description	9
5.1.2 Field Documentation	10
5.1.2.1 vernum	10
5.1.2.2 vertices	10
5.2 Vertex Struct Reference	10
5.2.1 Detailed Description	10
5.2.2 Field Documentation	11
5.2.2.1 next	11
5.2.2.2 number	11
6 File Documentation	13
6.1 graphs.c File Reference	13
6.1.1 Detailed Description	14
6.1.2 Function Documentation	14
6.1.2.1 add_edge()	14
6.1.2.2 add_vertex()	14
6.1.2.3 delete_graph()	14
6.1.2.4 print_graph()	15
6.1.2.5 remove_vertex()	15
6.1.2.6 upload_graph()	15
6.2 graphs.h File Reference	16
6.2.1 Detailed Description	16
6.2.2 Function Documentation	17
6.2.2.1 add_edge()	17
6.2.2.2 add_vertex()	17
6.2.2.3 delete_graph()	17
6.2.2.4 print_graph()	18
6.2.2.5 remove_vertex()	18
6.2.2.6 upload_graph()	18
Index	19

Chapter 1

README

Chapter 2

Todo List

Class `Graph`

maybe rework? the way to store a graph is too convoluted

Global `remove_vertex` (struct `Graph` *graph, int vertex)

this function is cringe

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

Graph	Structure of a graph	9
Vertex	Structure of a vertex l1 list	10

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

graphs.c	Graph implementation	13
graphs.h	Graph header	16

Chapter 5

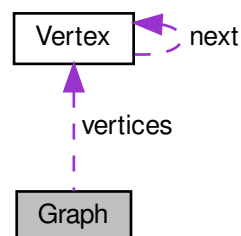
Data Structure Documentation

5.1 Graph Struct Reference

structure of a graph

```
#include <graphs.h>
```

Collaboration diagram for Graph:



Data Fields

- struct `Vertex` ** `vertices`
- int `vernum`

5.1.1 Detailed Description

structure of a graph

Todo maybe rework? the way to store a graph is too convoluted

5.1.2 Field Documentation

5.1.2.1 vernum

```
int Graph::vernum
```

store the amount of verices in a graph

5.1.2.2 vertices

```
struct Vertex** Graph::vertices
```

an array of l1 lists to store relations

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

- [graphs.h](#)

5.2 Vertex Struct Reference

structure of a vertex l1 list

```
#include <graphs.h>
```

Collaboration diagram for Vertex:



Data Fields

- int [number](#)
- struct [Vertex](#) * [next](#)

5.2.1 Detailed Description

structure of a vertex l1 list

5.2.2 Field Documentation

5.2.2.1 next

```
struct Vertex* Vertex::next
```

pointer to next in list structure

5.2.2.2 number

```
int Vertex::number
```

storing an end to an edge

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

- [graphs.h](#)

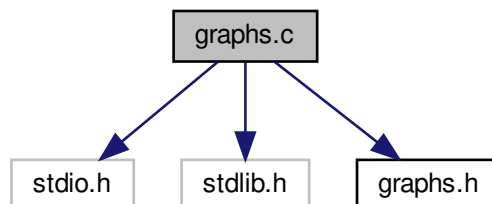
Chapter 6

File Documentation

6.1 graphs.c File Reference

graph implementation

```
#include <stdio.h>
#include <stdlib.h>
#include "graphs.h"
Include dependency graph for graphs.c:
```



Functions

- struct `Graph` `upload_graph` (char *filename)
initialize graph from file
- void `add_edge` (struct `Graph` *graph, int from, int to)
add a correlation between two vertices
- void `print_graph` (struct `Graph` graph)
print graph in a readable manner
- void `delete_graph` (struct `Graph` *graph)
delete graph and all of its members
- void `add_vertex` (struct `Graph` *graph)
add a vertex to a graph
- void `remove_vertex` (struct `Graph` *graph, int vertex)
remove a vertex from a graph

6.1.1 Detailed Description

graph implementation

6.1.2 Function Documentation

6.1.2.1 add_edge()

```
void add_edge (
    struct Graph * graph,
    int from,
    int to )
```

add a correlation between two vertices

Parameters

<i>graph</i>	which graph to add an edge
<i>from</i>	where the start of a vertex is
<i>to</i>	where the end of a vertex is

6.1.2.2 add_vertex()

```
void add_vertex (
    struct Graph * graph )
```

add a vertex to a graph

Parameters

<i>graph</i>	graph to which the vertex should be added
--------------	---

6.1.2.3 delete_graph()

```
void delete_graph (
    struct Graph * graph )
```

delete graph and all of its members

Parameters

<i>graph</i>	graph to delete
--------------	-----------------

6.1.2.4 print_graph()

```
void print_graph (
    struct Graph graph )
```

print graph in a readable manner

Parameters

<i>graph</i>	graph that needs to be printed
--------------	--------------------------------

6.1.2.5 remove_vertex()

```
void remove_vertex (
    struct Graph * graph,
    int vertex )
```

remove a vertex from a graph

Parameters

<i>graph</i>	graph from which the vertex should be removed
<i>vertex</i>	number of vertex that should be removed

Todo this function is cringe

6.1.2.6 upload_graph()

```
struct Graph upload_graph (
    char * filename )
```

initialize graph from file

Parameters

<i>filename</i>	name of the file from which to upload
-----------------	---------------------------------------

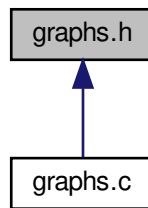
Returns

graph which the matrix determines

6.2 graphs.h File Reference

graph header

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [Graph](#)
structure of a graph
- struct [Vertex](#)
structure of a vertex l1 list

Functions

- struct [Graph](#) [upload_graph](#) (char *filename)
initialize graph from file
- void [print_graph](#) (struct [Graph](#) graph)
print graph in a readable manner
- void [add_edge](#) (struct [Graph](#) *graph, int from, int to)
add a correlation between two vertices
- void [delete_graph](#) (struct [Graph](#) *graph)
delete graph and all of its members
- void [add_vertex](#) (struct [Graph](#) *graph)
add a vertex to a graph
- void [remove_vertex](#) (struct [Graph](#) *graph, int vertex)
remove a vertex from a graph

6.2.1 Detailed Description

graph header

6.2.2 Function Documentation

6.2.2.1 add_edge()

```
void add_edge (
    struct Graph * graph,
    int from,
    int to )
```

add a correlation between two vertices

Parameters

<i>graph</i>	which graph to add an edge
<i>from</i>	where the start of a vertex is
<i>to</i>	where the end of a vertex is

6.2.2.2 add_vertex()

```
void add_vertex (
    struct Graph * graph )
```

add a vertex to a graph

Parameters

<i>graph</i>	graph to which the vertex should be added
--------------	---

6.2.2.3 delete_graph()

```
void delete_graph (
    struct Graph * graph )
```

delete graph and all of its members

Parameters

<i>graph</i>	graph to delete
--------------	-----------------

6.2.2.4 print_graph()

```
void print_graph (
    struct Graph graph )
```

print graph in a readable manner

Parameters

<i>graph</i>	graph that needs to be printed
--------------	--------------------------------

6.2.2.5 remove_vertex()

```
void remove_vertex (
    struct Graph * graph,
    int vertex )
```

remove a vertex from a graph

Parameters

<i>graph</i>	graph from which the vertex should be removed
<i>vertex</i>	number of vertex that should be removed

Todo this function is cringe

6.2.2.6 upload_graph()

```
struct Graph upload_graph (
    char * filename )
```

initialize graph from file

Parameters

<i>filename</i>	name of the file from which to upload
-----------------	---------------------------------------

Returns

graph which the matrix determines

Index

- add_edge
 - graphs.c, [14](#)
 - graphs.h, [17](#)
- add_vertex
 - graphs.c, [14](#)
 - graphs.h, [17](#)
- delete_graph
 - graphs.c, [14](#)
 - graphs.h, [17](#)
- Graph, [9](#)
 - vernum, [10](#)
 - vertices, [10](#)
- graphs.c, [13](#)
 - add_edge, [14](#)
 - add_vertex, [14](#)
 - delete_graph, [14](#)
 - print_graph, [15](#)
 - remove_vertex, [15](#)
 - upload_graph, [15](#)
- graphs.h, [16](#)
 - add_edge, [17](#)
 - add_vertex, [17](#)
 - delete_graph, [17](#)
 - print_graph, [17](#)
 - remove_vertex, [18](#)
 - upload_graph, [18](#)
- next
 - Vertex, [11](#)
- number
 - Vertex, [11](#)
- print_graph
 - graphs.c, [15](#)
 - graphs.h, [17](#)
- remove_vertex
 - graphs.c, [15](#)
 - graphs.h, [18](#)
- upload_graph
 - graphs.c, [15](#)
 - graphs.h, [18](#)
- vernum
 - Graph, [10](#)
- Vertex, [10](#)
 - next, [11](#)
 - number, [11](#)
- vertices
 - Graph, [10](#)