

Lab 12 Graph - Ernest Landrito

Generated by Doxygen 1.8.5

Wed Nov 27 2013 01:23:44

Contents

| | | |
|----------|--|----------|
| 1 | Class Index | 1 |
| 1.1 | Class List | 1 |
| 2 | File Index | 3 |
| 2.1 | File List | 3 |
| 3 | Class Documentation | 5 |
| 3.1 | WeightedGraph::Vertex Class Reference | 5 |
| 3.1.1 | Member Function Documentation | 5 |
| 3.1.1.1 | getColor | 5 |
| 3.1.1.2 | getLabel | 5 |
| 3.1.1.3 | setColor | 5 |
| 3.1.1.4 | setLabel | 5 |
| 3.2 | Vertex Class Reference | 5 |
| 3.2.1 | Member Data Documentation | 6 |
| 3.2.1.1 | color | 6 |
| 3.2.1.2 | label | 6 |
| 3.3 | WeightedGraph Class Reference | 6 |
| 3.3.1 | Constructor & Destructor Documentation | 7 |
| 3.3.1.1 | WeightedGraph | 7 |
| 3.3.1.2 | WeightedGraph | 7 |
| 3.3.1.3 | ~WeightedGraph | 8 |
| 3.3.1.4 | WeightedGraph | 8 |
| 3.3.1.5 | WeightedGraph | 8 |
| 3.3.1.6 | ~WeightedGraph | 8 |
| 3.3.2 | Member Function Documentation | 8 |
| 3.3.2.1 | areAllEven | 8 |
| 3.3.2.2 | clear | 8 |
| 3.3.2.3 | clear | 8 |
| 3.3.2.4 | computePaths | 9 |
| 3.3.2.5 | edgeWeight | 9 |
| 3.3.2.6 | getEdgeWeight | 9 |

| | | |
|----------|--|----|
| 3.3.2.7 | getEdgeWeight | 9 |
| 3.3.2.8 | hasProperColoring | 9 |
| 3.3.2.9 | insertEdge | 9 |
| 3.3.2.10 | insertEdge | 9 |
| 3.3.2.11 | insertVertex | 10 |
| 3.3.2.12 | insertVertex | 10 |
| 3.3.2.13 | isEmpty | 10 |
| 3.3.2.14 | isEmpty | 10 |
| 3.3.2.15 | isFull | 11 |
| 3.3.2.16 | isFull | 11 |
| 3.3.2.17 | operator= | 11 |
| 3.3.2.18 | operator= | 11 |
| 3.3.2.19 | removeEdge | 12 |
| 3.3.2.20 | removeEdge | 12 |
| 3.3.2.21 | removeVertex | 12 |
| 3.3.2.22 | removeVertex | 12 |
| 3.3.2.23 | retrieveVertex | 13 |
| 3.3.2.24 | retrieveVertex | 13 |
| 3.3.2.25 | showShortestPaths | 13 |
| 3.3.2.26 | showStructure | 13 |
| 3.3.2.27 | showStructure | 13 |
| 3.3.3 | Member Data Documentation | 13 |
| 3.3.3.1 | DEF_MAX_GRAPH_SIZE | 14 |
| 3.3.3.2 | INFINITE_EDGE_WT | 14 |
| 3.3.3.3 | MAX_GRAPH_SIZE | 14 |
| 3.3.3.4 | VERTEX_LABEL_LENGTH | 14 |
| 3.4 | WtGraph Class Reference | 14 |
| 3.4.1 | Constructor & Destructor Documentation | 14 |
| 3.4.1.1 | ~WtGraph | 14 |
| 3.4.2 | Member Function Documentation | 14 |
| 3.4.2.1 | clear | 14 |
| 3.4.2.2 | edgeWeight | 14 |
| 3.4.2.3 | getEdgeWeight | 14 |
| 3.4.2.4 | hasProperColoring | 14 |
| 3.4.2.5 | insertEdge | 14 |
| 3.4.2.6 | insertVertex | 15 |
| 3.4.2.7 | isEmpty | 15 |
| 3.4.2.8 | isFull | 15 |
| 3.4.2.9 | removeEdge | 15 |
| 3.4.2.10 | removeVertex | 15 |

| | | |
|--------------|----------------------------------|-----------|
| 3.4.2.11 | retrieveVertex | 15 |
| 3.4.2.12 | showStructure | 15 |
| 3.4.2.13 | throw | 15 |
| 4 | File Documentation | 17 |
| 4.1 | config.h File Reference | 17 |
| 4.1.1 | Macro Definition Documentation | 17 |
| 4.1.1.1 | LAB12_TEST1 | 17 |
| 4.1.1.2 | LAB12_TEST2 | 17 |
| 4.1.1.3 | LAB12_TEST3 | 17 |
| 4.2 | show12.cpp File Reference | 17 |
| 4.3 | test12.cpp File Reference | 17 |
| 4.3.1 | Function Documentation | 18 |
| 4.3.1.1 | main | 18 |
| 4.3.1.2 | print_help | 18 |
| 4.4 | WeightedGraph.cpp File Reference | 18 |
| 4.5 | WeightedGraph.cs File Reference | 18 |
| 4.6 | WeightedGraph.h File Reference | 18 |
| 4.7 | WeightedGraph2.h File Reference | 18 |
| 4.8 | WeightedGraph3.h File Reference | 19 |
| 4.8.1 | Variable Documentation | 19 |
| 4.8.1.1 | defMaxGraphSize | 19 |
| 4.8.1.2 | infiniteEdgeWt | 19 |
| 4.8.1.3 | vertexLabelLength | 19 |
| Index | | 20 |

Chapter 1

Class Index

1.1 Class List

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

| | |
|---------------------------------------|----|
| WeightedGraph::Vertex | 5 |
| Vertex | 5 |
| WeightedGraph | 6 |
| WtGraph | 14 |

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

| | |
|-----------------------------------|----|
| config.h | 17 |
| show12.cpp | 17 |
| test12.cpp | 17 |
| WeightedGraph.cpp | 18 |
| WeightedGraph.cs | 18 |
| WeightedGraph.h | 18 |
| WeightedGraph2.h | 18 |
| WeightedGraph3.h | 19 |

Chapter 3

Class Documentation

3.1 WeightedGraph::Vertex Class Reference

```
#include <WeightedGraph.h>
```

Public Member Functions

- void [setLabel](#) (const string &newLabel)
- string [getLabel](#) () const
- void [setColor](#) (char newColor)
- char [getColor](#) () const

3.1.1 Member Function Documentation

3.1.1.1 char WeightedGraph::Vertex::getColor () const [inline]

3.1.1.2 string WeightedGraph::Vertex::getLabel () const [inline]

3.1.1.3 void WeightedGraph::Vertex::setColor (char *newColor*) [inline]

3.1.1.4 void WeightedGraph::Vertex::setLabel (const string & *newLabel*) [inline]

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

- [WeightedGraph.h](#)

3.2 Vertex Class Reference

```
#include <WeightedGraph2.h>
```

Public Attributes

- char [label](#) [[vertexLabelLength](#)]
- char [color](#)

3.2.1 Member Data Documentation

3.2.1.1 `char Vertex::color`

3.2.1.2 `char Vertex::label`

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

- [WeightedGraph2.h](#)
- [WeightedGraph3.h](#)

3.3 WeightedGraph Class Reference

```
#include <WeightedGraph.h>
```

Classes

- class [Vertex](#)

Public Member Functions

- [WeightedGraph](#) (int maxNumber=[MAX_GRAPH_SIZE](#))
- [WeightedGraph](#) (const [WeightedGraph](#) &other)
- [WeightedGraph](#) & [operator=](#) (const [WeightedGraph](#) &other)
- [~WeightedGraph](#) ()
- void [insertVertex](#) (const [Vertex](#) &newVertex) throw ([logic_error](#))
- void [insertEdge](#) (const string &v1, const string &v2, int wt) throw ([logic_error](#))
- bool [retrieveVertex](#) (const string &v, [Vertex](#) &vData) const
- bool [getEdgeWeight](#) (const string &v1, const string &v2, int &wt) const throw ([logic_error](#))
- void [removeVertex](#) (const string &v) throw ([logic_error](#))
- void [removeEdge](#) (const string &v1, const string &v2) throw ([logic_error](#))
- void [clear](#) ()
- bool [isEmpty](#) () const
- bool [isFull](#) () const
- void [showStructure](#) () const
- void [showShortestPaths](#) () const
- bool [hasProperColoring](#) () const
- bool [areAllEven](#) () const
- [WeightedGraph](#) (int maxNumber=[defMaxGraphSize](#))
- [WeightedGraph](#) (const [WeightedGraph](#) &other)
- [WeightedGraph](#) & [operator=](#) (const [WeightedGraph](#) &other)
- [~WeightedGraph](#) ()
- void [insertVertex](#) ([Vertex](#) newVertex) throw ([logic_error](#))
- void [insertEdge](#) (char *v1, char *v2, int wt) throw ([logic_error](#))
- bool [retrieveVertex](#) (char *v, [Vertex](#) &vData) const
- int [edgeWeight](#) (char *v1, char *v2, int &wt) const throw ([logic_error](#))
- bool [getEdgeWeight](#) (char *v1, char *v2, int &wt) const throw ([logic_error](#))
- void [removeVertex](#) (char *v) throw ([logic_error](#))
- void [removeEdge](#) (char *v1, char *v2) throw ([logic_error](#))
- void [clear](#) ()
- void [computePaths](#) ()
- bool [isEmpty](#) () const
- bool [isFull](#) () const
- void [showStructure](#) () const

Static Public Attributes

- static const int `MAX_GRAPH_SIZE` = 10
- static const int `INFINITE_EDGE_WT` = INT_MAX
- static const int `DEF_MAX_GRAPH_SIZE` = 10
- static const int `VERTEX_LABEL_LENGTH` = 11

3.3.1 Constructor & Destructor Documentation

3.3.1.1 `WeightedGraph::WeightedGraph (int maxNumber = MAX_GRAPH_SIZE)`

Precondition

New Graph Class

Postcondition

dynamically allocated array

Parameters

| | |
|------------------|------------------------|
| <i>maxNumber</i> | the number of vertexes |
|------------------|------------------------|

Algorithm:

- set size to zero
- set maxSize
- allocate arrays

Exceptional/Error Conditions:

- none

3.3.1.2 `WeightedGraph::WeightedGraph (const WeightedGraph & other)`

Precondition

new graph class

Postcondition

a deep copy of source graph

Parameters

| | |
|--------------|--------------------------------|
| <i>other</i> | Source graph to be deep copied |
|--------------|--------------------------------|

Algorithm:

- Use overloaded assignment operator to copy the data from the source to this class

3.3.1.3 `WeightedGraph::~~WeightedGraph ()`

Precondition

a Graph

Postcondition

deallocated Heap

Algorithm:

- delete the arrays

3.3.1.4 `WeightedGraph::WeightedGraph (int maxNumber = defMaxGraphSize)`

3.3.1.5 `WeightedGraph::WeightedGraph (const WeightedGraph & other)`

3.3.1.6 `WeightedGraph::~~WeightedGraph ()`

3.3.2 Member Function Documentation

3.3.2.1 `bool WeightedGraph::areAllEven () const`

Precondition

a graph

Postcondition

edge removed from the edge matrix

Algorithm:

- count the number of edges per vertex
- if the number of edges isnt even, return false
- else return true

3.3.2.2 `void WeightedGraph::clear ()`

3.3.2.3 `void WeightedGraph::clear ()`

Precondition

a graph

Postcondition

an empty graph

Algorithm:

- set the size of the heap to 0;

3.3.2.4 void WeightedGraph::computePaths ()

3.3.2.5 int WeightedGraph::edgeWeight (char * v1, char * v2, int & wt) const throw logic_error)

3.3.2.6 bool WeightedGraph::getEdgeWeight (char * v1, char * v2, int & wt) const throw logic_error)

3.3.2.7 bool WeightedGraph::getEdgeWeight (const string & v1, const string & v2, int & wt) const throw logic_error)

Precondition

a graph

Postcondition

weight data in wt

Parameters

| | |
|-----------|----------------------------|
| <i>v1</i> | string to be searched for |
| <i>v2</i> | string to be searched for |
| <i>wt</i> | weight data to be returned |

Returns

Return if successful

Exceptions

| | |
|--------------------|---------------------------------|
| <i>logic_error</i> | Throw if vertex's are not found |
|--------------------|---------------------------------|

Algorithm:

- searches for the index locations of the vertexs
- if the index locations is found set wt
- else return false

3.3.2.8 bool WeightedGraph::hasProperColoring () const

Precondition

a graph

Postcondition

edge removed from the edge matrix

Algorithm:

- search if any vertex has the same color as another

3.3.2.9 void WeightedGraph::insertEdge (char * v1, char * v2, int wt) throw logic_error)

3.3.2.10 void WeightedGraph::insertEdge (const string & v1, const string & v2, int wt) throw logic_error)

Precondition

a graph

Postcondition

Edge inserted into the edge matrix

Parameters

| | |
|-----------|---------------------------|
| <i>v1</i> | First vertex to be found |
| <i>v2</i> | Second vertex to be found |

Exceptions

| | |
|--------------------|---|
| <i>logic_error</i> | Throws an error if vertex's arent found |
|--------------------|---|

Algorithm:

- searches for the index locations of the vertex's
- if the index locations are not found throw logic error
- else set the edge

3.3.2.11 void WeightedGraph::insertVertex (Vertex *newVertex*) throw logic_error)

3.3.2.12 void WeightedGraph::insertVertex (const Vertex & *newVertex*) throw logic_error)

Precondition

a Graph

Postcondition

a new vertex inserted into the graph

Parameters

| | |
|------------------|-----------------------------------|
| <i>newVertex</i> | Data to be inserted into the Heap |
|------------------|-----------------------------------|

Exceptions

| | |
|--------------------|------------------------------------|
| <i>logic_error</i> | Throws error if theg graph is full |
|--------------------|------------------------------------|

Algorithm:

- see if the vertex is in the array if so update and return
- check if the graph is full if so throw
- else insert the vertex and increase size

3.3.2.13 bool WeightedGraph::isEmpty () const

3.3.2.14 bool WeightedGraph::isEmpty () const

Precondition

a Graph

Postcondition

returns if the Graph is empty

Returns

returns if the Graph is empty

Algorithm:

- return if the size is equal to 0

3.3.2.15 `bool WeightedGraph::isFull () const`

3.3.2.16 `bool WeightedGraph::isFull () const`

Precondition

a graph

Postcondition

returns if the graph is full

Returns

returns if the graph is full

Algorithm:

- return if the size is equal to the max size

3.3.2.17 `WeightedGraph& WeightedGraph::operator= (const WeightedGraph & other)`

3.3.2.18 `WeightedGraph & WeightedGraph::operator= (const WeightedGraph & other)`

Precondition

a Graph class

Postcondition

a deep copy of source Graph

Parameters

| | |
|---------------|--------------------------------|
| <i>source</i> | Source graph to be deep copied |
|---------------|--------------------------------|

Algorithm:

- if the heap is not empty delete teh matrices
- and it is itself return itself
- allocate new arrays
- copy the data
- return this

3.3.2.19 void WeightedGraph::removeEdge (char * v1, char * v2) throw logic_error)

3.3.2.20 void WeightedGraph::removeEdge (const string & v1, const string & v2) throw logic_error)

Precondition

a graph

Postcondition

edge removed from the edge matrix

Parameters

| | |
|----|---------------------------|
| v1 | vertex to be searched for |
| v2 | vertex to be searched for |

Exceptions

| | |
|--------------------|--------------------------------|
| <i>logic_error</i> | Throw if vertexs are not found |
|--------------------|--------------------------------|

Algorithm:

- searches for the index locations of the vertex
- set edge locations to infinite edge wt
- throw logic error if vertex's arent found

3.3.2.21 void WeightedGraph::removeVertex (char * v) throw logic_error)

3.3.2.22 void WeightedGraph::removeVertex (const string & v) throw logic_error)

Precondition

a graph

Postcondition

vertex removed

Parameters

| | |
|---|----------------------|
| v | vertex to be removed |
|---|----------------------|

Exceptions

| | |
|--------------------|------------------------------|
| <i>logic_error</i> | Throw if vertex is not found |
|--------------------|------------------------------|

Algorithm:

- searches for the index locations of the vertex
- shift the adj matrix rows left and columns up
- shift vertices left

3.3.2.23 `bool WeightedGraph::retrieveVertex (char * v, Vertex & vData) const`

3.3.2.24 `bool WeightedGraph::retrieveVertex (const string & v, Vertex & vData) const`

Precondition

a graph

Postcondition

vertex data in vData

Parameters

| | |
|--------------|----------------------------|
| <i>v</i> | string to be searched for |
| <i>vData</i> | vertex data to be returned |

Returns

Return if successful

Algorithm:

- searches for the index location of the vertex
- if the index locations is found set vData
- else return false

3.3.2.25 `void WeightedGraph::showShortestPaths () const`

Precondition

a graph

Postcondition

paths changed to shortest paths

Algorithm:

- Initialize path array
- if there is a path between j and m
- and there is a path between m and k
- andt the sum of j,m and m,k is less than j,k
- replace j,k with j,m + m,k

3.3.2.26 `void WeightedGraph::showStructure () const`

3.3.2.27 `void WeightedGraph::showStructure () const`

3.3.3 Member Data Documentation

3.3.3.1 `const int WeightedGraph::DEF_MAX_GRAPH_SIZE = 10` `[static]`

3.3.3.2 `static const int WeightedGraph::INFINITE_EDGE_WT = INT_MAX` `[static]`

3.3.3.3 `const int WeightedGraph::MAX_GRAPH_SIZE = 10` `[static]`

3.3.3.4 `const int WeightedGraph::VERTEX_LABEL_LENGTH = 11` `[static]`

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

- [WeightedGraph.h](#)
- [WeightedGraph2.h](#)
- [show12.cpp](#)
- [WeightedGraph.cpp](#)

3.4 WtGraph Class Reference

```
#include <WeightedGraph3.h>
```

Public Member Functions

- [throw](#) (bad_alloc)
- [~WtGraph](#) ()
- void [insertVertex](#) ([Vertex](#) newVertex) throw (logic_error)
- void [insertEdge](#) (char *v1, char *v2, int wt) throw (logic_error)
- bool [retrieveVertex](#) (char *v, [Vertex](#) &vData) const
- bool [edgeWeight](#) (char *v1, char *v2, int &wt) const throw (logic_error)
- bool [getEdgeWeight](#) (char *v1, char *v2, int &wt) const throw (logic_error)
- void [removeVertex](#) (char *v) throw (logic_error)
- void [removeEdge](#) (char *v1, char *v2) throw (logic_error)
- void [clear](#) ()
- bool [isEmpty](#) () const
- bool [isFull](#) () const
- bool [hasProperColoring](#) () const
- void [showStructure](#) () const

3.4.1 Constructor & Destructor Documentation

3.4.1.1 `WtGraph::~~WtGraph ()`

3.4.2 Member Function Documentation

3.4.2.1 `void WtGraph::clear ()`

3.4.2.2 `bool WtGraph::edgeWeight (char * v1, char * v2, int & wt) const throw logic_error)`

3.4.2.3 `bool WtGraph::getEdgeWeight (char * v1, char * v2, int & wt) const throw logic_error)`

3.4.2.4 `bool WtGraph::hasProperColoring () const`

3.4.2.5 `void WtGraph::insertEdge (char * v1, char * v2, int wt) throw logic_error)`

- 3.4.2.6 void WtGraph::insertVertex (*Vertex newVertex*) throw logic_error)
- 3.4.2.7 bool WtGraph::isEmpty () const
- 3.4.2.8 bool WtGraph::isFull () const
- 3.4.2.9 void WtGraph::removeEdge (char * *v1*, char * *v2*) throw logic_error)
- 3.4.2.10 void WtGraph::removeVertex (char * *v*) throw logic_error)
- 3.4.2.11 bool WtGraph::retrieveVertex (char * *v*, Vertex & *vData*) const
- 3.4.2.12 void WtGraph::showStructure () const
- 3.4.2.13 WtGraph::throw (bad_alloc)

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

- [WeightedGraph3.h](#)
- [WeightedGraph.cs](#)

Chapter 4

File Documentation

4.1 config.h File Reference

Macros

- `#define LAB12_TEST1 1`
- `#define LAB12_TEST2 1`
- `#define LAB12_TEST3 1`

4.1.1 Macro Definition Documentation

4.1.1.1 `#define LAB12_TEST1 1`

`WeightedGraph` class configuration file. Activate test #N by defining the corresponding `LAB12_TESTN` to have the value 1.

4.1.1.2 `#define LAB12_TEST2 1`

4.1.1.3 `#define LAB12_TEST3 1`

4.2 show12.cpp File Reference

4.3 test12.cpp File Reference

```
#include <iostream>
#include <cstring>
#include <cctype>
#include "WeightedGraph.h"
#include "config.h"
```

Functions

- void `print_help` ()
- int `main` ()

4.3.1 Function Documentation

4.3.1.1 `int main ()`

4.3.1.2 `void print_help ()`

4.4 WeightedGraph.cpp File Reference

```
#include <stdexcept>
#include <iostream>
#include <climits>
#include <string>
#include "WeightedGraph.h"
```

4.5 WeightedGraph.cs File Reference

```
#include <iostream>
#include <cstring>
#include "wtgraph.h"
```

4.6 WeightedGraph.h File Reference

```
#include <stdexcept>
#include <iostream>
#include <climits>
#include <string>
```

Classes

- class [WeightedGraph](#)
- class [WeightedGraph::Vertex](#)

4.7 WeightedGraph2.h File Reference

```
#include <climits>
#include <new>
#include <stdexcept>
```

Classes

- class [Vertex](#)
- class [WeightedGraph](#)

4.8 WeightedGraph3.h File Reference

```
#include <climits>
#include <new>
#include <stdexcept>
```

Classes

- class [Vertex](#)
- class [WtGraph](#)

Variables

- const int [defMaxGraphSize](#) = 10
- const int [vertexLabelLength](#) = 11
- const int [infiniteEdgeWt](#) = INT_MAX

4.8.1 Variable Documentation

4.8.1.1 `const int defMaxGraphSize = 10`

4.8.1.2 `const int infiniteEdgeWt = INT_MAX`

4.8.1.3 `const int vertexLabelLength = 11`

Index

- ~WeightedGraph
 - WeightedGraph, [7](#), [8](#)
- ~WtGraph
 - WtGraph, [14](#)
- areAllEven
 - WeightedGraph, [8](#)
- clear
 - WeightedGraph, [8](#)
 - WtGraph, [14](#)
- color
 - Vertex, [6](#)
- computePaths
 - WeightedGraph, [8](#)
- config.h, [17](#)
 - LAB12_TEST1, [17](#)
 - LAB12_TEST2, [17](#)
 - LAB12_TEST3, [17](#)
- DEF_MAX_GRAPH_SIZE
 - WeightedGraph, [13](#)
- defMaxGraphSize
 - WeightedGraph3.h, [19](#)
- edgeWeight
 - WeightedGraph, [9](#)
 - WtGraph, [14](#)
- getColor
 - WeightedGraph::Vertex, [5](#)
- getEdgeWeight
 - WeightedGraph, [9](#)
 - WtGraph, [14](#)
- getLabel
 - WeightedGraph::Vertex, [5](#)
- hasProperColoring
 - WeightedGraph, [9](#)
 - WtGraph, [14](#)
- INFINITE_EDGE_WT
 - WeightedGraph, [14](#)
- infiniteEdgeWt
 - WeightedGraph3.h, [19](#)
- insertEdge
 - WeightedGraph, [9](#)
 - WtGraph, [14](#)
- insertVertex
 - WeightedGraph, [10](#)
 - WtGraph, [14](#)
- isEmpty
 - WeightedGraph, [10](#)
 - WtGraph, [15](#)
- isFull
 - WeightedGraph, [11](#)
 - WtGraph, [15](#)
- LAB12_TEST1
 - config.h, [17](#)
- LAB12_TEST2
 - config.h, [17](#)
- LAB12_TEST3
 - config.h, [17](#)
- label
 - Vertex, [6](#)
- MAX_GRAPH_SIZE
 - WeightedGraph, [14](#)
- main
 - test12.cpp, [18](#)
- operator=
 - WeightedGraph, [11](#)
- print_help
 - test12.cpp, [18](#)
- removeEdge
 - WeightedGraph, [11](#), [12](#)
 - WtGraph, [15](#)
- removeVertex
 - WeightedGraph, [12](#)
 - WtGraph, [15](#)
- retrieveVertex
 - WeightedGraph, [12](#), [13](#)
 - WtGraph, [15](#)
- setColor
 - WeightedGraph::Vertex, [5](#)
- setLabel
 - WeightedGraph::Vertex, [5](#)
- show12.cpp, [17](#)
- showShortestPaths
 - WeightedGraph, [13](#)
- showStructure
 - WeightedGraph, [13](#)
 - WtGraph, [15](#)
- test12.cpp, [17](#)
 - main, [18](#)
 - print_help, [18](#)

- throw
 - WtGraph, 15
- VERTEX_LABEL_LENGTH
 - WeightedGraph, 14
- Vertex, 5
 - color, 6
 - label, 6
- vertexLabelLength
 - WeightedGraph3.h, 19
- WeightedGraph, 6
 - ~WeightedGraph, 7, 8
 - areAllEven, 8
 - clear, 8
 - computePaths, 8
 - DEF_MAX_GRAPH_SIZE, 13
 - edgeWeight, 9
 - getEdgeWeight, 9
 - hasProperColoring, 9
 - INFINITE_EDGE_WT, 14
 - insertEdge, 9
 - insertVertex, 10
 - isEmpty, 10
 - isFull, 11
 - MAX_GRAPH_SIZE, 14
 - operator=, 11
 - removeEdge, 11, 12
 - removeVertex, 12
 - retrieveVertex, 12, 13
 - showShortestPaths, 13
 - showStructure, 13
 - VERTEX_LABEL_LENGTH, 14
 - WeightedGraph, 7, 8
 - WeightedGraph, 7, 8
- WeightedGraph.cpp, 18
- WeightedGraph.cs, 18
- WeightedGraph.h, 18
- WeightedGraph2.h, 18
- WeightedGraph3.h, 19
 - defMaxGraphSize, 19
 - infiniteEdgeWt, 19
 - vertexLabelLength, 19
- WeightedGraph::Vertex, 5
 - getColor, 5
 - getLabel, 5
 - setColor, 5
 - setLabel, 5
- WtGraph, 14
 - ~WtGraph, 14
 - clear, 14
 - edgeWeight, 14
 - getEdgeWeight, 14
 - hasProperColoring, 14
 - insertEdge, 14
 - insertVertex, 14
 - isEmpty, 15
 - isFull, 15
 - removeEdge, 15
 - removeVertex, 15
 - retrieveVertex, 15
 - showStructure, 15
 - throw, 15