

graphs

0.1.0

Generated by Doxygen 1.8.17

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 Graph Class Reference	5
3.1.1 Detailed Description	5
3.1.2 Constructor & Destructor Documentation	5
3.1.2.1 Graph()	5
3.1.3 Member Function Documentation	6
3.1.3.1 addEdge()	6
3.1.3.2 hasEdge()	6
3.1.3.3 inEdges()	6
3.1.3.4 nVertices()	6
3.1.3.5 outEdges()	7
3.1.3.6 printgraph()	7
3.1.3.7 removeEdge()	7
4 File Documentation	9
4.1 /home/brandon/CPTR227/HW14/graphs/src/main.cpp File Reference	9
4.1.1 Detailed Description	10
4.1.2 Function Documentation	10
4.1.2.1 bfs()	10
4.1.2.2 main()	10
Index	13

Chapter 1

Class Index

1.1 Class List

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

Graph	5
---------------------------------	---

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

/home/brandon/CPTR227/HW14/graphs/src/main.cpp	
This is HW14 - Matrix Based Graph for CPTR 227 class	9

Chapter 3

Class Documentation

3.1 Graph Class Reference

Public Member Functions

- [Graph](#) (int nV)
- void [addEdge](#) (int i, int j)
- void [removeEdge](#) (int i, int j)
- bool [hasEdge](#) (int i, int j)
- void [outEdges](#) (int i, vector< bool > &edges)
- void [inEdges](#) (int i, vector< bool > &edges)
- int [nVertices](#) ()
- void [printgraph](#) ()

3.1.1 Detailed Description

Definition at line 14 of file main.cpp.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Graph()

```
Graph::Graph (  
    int nV ) [inline]
```

Definition at line 19 of file main.cpp.

```
19         {  
20             n = nV;  
21             for(int i = 0; i < n; i++) {  
22                 vector<bool> newRow(n, false);  
23                 a.push_back(newRow);  
24             }  
25         }
```

3.1.3 Member Function Documentation

3.1.3.1 addEdge()

```
void Graph::addEdge (
    int i,
    int j ) [inline]
```

Definition at line 27 of file main.cpp.

```
27 {
28     a[i][j] = true;
29 }
```

3.1.3.2 hasEdge()

```
bool Graph::hasEdge (
    int i,
    int j ) [inline]
```

Definition at line 35 of file main.cpp.

```
35 {
36     return a[i][j];
37 }
```

3.1.3.3 inEdges()

```
void Graph::inEdges (
    int i,
    vector< bool > & edges ) [inline]
```

Definition at line 44 of file main.cpp.

```
44 {
45     for (int j = 0; j < n; j++)
46         if (a[j][i]) edges.push_back(j);
47 }
```

3.1.3.4 nVertices()

```
int Graph::nVertices ( ) [inline]
```

Definition at line 49 of file main.cpp.

```
49 {
50     return n;
51 }
```

3.1.3.5 outEdges()

```
void Graph::outEdges (
    int i,
    vector< bool > & edges ) [inline]
```

Definition at line 39 of file main.cpp.

```
39 {
40     for (int j = 0; j < n; j++)
41         if (a[i][j]) edges.push_back(j);
42 }
```

3.1.3.6 printgraph()

```
void Graph::printgraph ( ) [inline]
```

Definition at line 53 of file main.cpp.

```
53 {
54     cout << "\t";
55     for(int i = 0; i < n; i++) {
56         cout << i << "\t";
57     }
58     cout << endl;
59     for(int i = 0; i < n; i++) {
60         cout << i << "\t";
61         for(int j = 0; j < n; j++)
62             cout << a[i][j] << "\t";
63         cout << endl;
64     }
65 }
```

3.1.3.7 removeEdge()

```
void Graph::removeEdge (
    int i,
    int j ) [inline]
```

Definition at line 31 of file main.cpp.

```
31 {
32     a[i][j] = false;
33 }
```

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

- </home/brandon/CPTR227/HW14/graphs/src/main.cpp>

Chapter 4

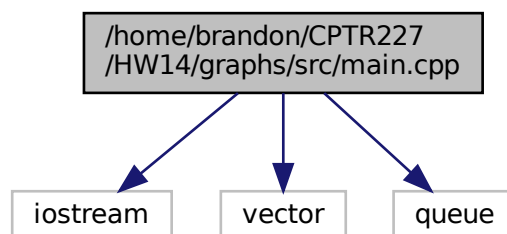
File Documentation

4.1 /home/brandon/CPTR227/HW14/graphs/src/main.cpp File Reference

This is HW14 - Matrix Based [Graph](#) for CPTR 227 class.

```
#include <iostream>
#include <vector>
#include <queue>
```

Include dependency graph for main.cpp:



Classes

- class [Graph](#)

Functions

- void [bfs](#) ([Graph](#) &g, int r)
- int [main](#) ()

4.1.1 Detailed Description

This is HW14 - Matrix Based [Graph](#) for CPTR 227 class.

Author

Brandon Yi

Date

4/7/2021

4.1.2 Function Documentation

4.1.2.1 bfs()

```
void bfs (
    Graph & g,
    int r )
```

Definition at line 70 of file main.cpp.

```
70     {
71         bool *seen = new bool[g.nVertices()];
72         queue <int> q;
73         q.push(r);
74         seen[r] = true;
75         while (q.size() > 0) {
76             int i = q.front();
77             q.pop();
78             vector<bool> edges;
79             g.outEdges(i, edges);
80             for (int k = 0; k < edges.size(); k++) {
81                 int j = edges.at(k);
82                 if (!seen[j]) {
83                     q.push(j);
84                     seen[j] = true;
85                 }
86             }
87         }
88         delete[] seen;
89     }
```

4.1.2.2 main()

```
int main ( )
```

Definition at line 91 of file main.cpp.

```
91     {
92         Graph g(12);
93         g.addEdge(0,1);
94         g.addEdge(1,0);
95         g.addEdge(1,2);
96         g.addEdge(2,1);
97         g.addEdge(2,3);
98         g.addEdge(3,2);
99         g.addEdge(0,4);
100        g.addEdge(4,0);
101        g.addEdge(4,5);
```

```
102     g.addEdge(5, 4);
103     g.addEdge(5, 6);
104     g.addEdge(6, 5);
105     g.addEdge(6, 7);
106     g.addEdge(7, 6);
107     g.addEdge(3, 7);
108     g.addEdge(7, 3);
109     g.addEdge(1, 5);
110     g.addEdge(5, 1);
111     g.addEdge(2, 6);
112     g.addEdge(6, 2);
113     g.addEdge(4, 8);
114     g.addEdge(8, 4);
115     g.addEdge(8, 9);
116     g.addEdge(9, 8);
117     g.addEdge(5, 9);
118     g.addEdge(9, 5);
119     g.addEdge(9, 10);
120     g.addEdge(10, 9);
121     g.addEdge(6, 10);
122     g.addEdge(10, 6);
123     g.addEdge(10, 11);
124     g.addEdge(11, 10);
125     g.addEdge(7, 11);
126     g.addEdge(11, 7);
127     g.addEdge(1, 6);
128     g.addEdge(5, 2);
129     g.printgraph();
130 }
```


Index

/home/brandon/CPTR227/HW14/graphs/src/main.cpp,
[9](#)

addEdge
 Graph, [6](#)

bfs
 main.cpp, [10](#)

Graph, [5](#)
 addEdge, [6](#)
 Graph, [5](#)
 hasEdge, [6](#)
 inEdges, [6](#)
 nVertices, [6](#)
 outEdges, [6](#)
 printgraph, [7](#)
 removeEdge, [7](#)

hasEdge
 Graph, [6](#)

inEdges
 Graph, [6](#)

main
 main.cpp, [10](#)

main.cpp
 bfs, [10](#)
 main, [10](#)

nVertices
 Graph, [6](#)

outEdges
 Graph, [6](#)

printgraph
 Graph, [7](#)

removeEdge
 Graph, [7](#)