

John Plunk
Sec 503

Type of sources:	
People	Samia Kubar, Andrew Lam
Web Material (give URL)	Ecampus, Piazza
Printed Material	“Data Structures and Algorithms in C++” by M. Goodrich, R. Tamassia and D. Mount, 2011
Other Sources	

I certify that I have listed all the sources that I used to develop solutions to the submitted project report and code.

1. Your signature *John Plunk*

Typed Name: **John Plunk**

Date:

4-28-19

How To Run

To run this program, one must download the files and compile with the following command:

```
c++ -std=c++11 *.cpp
```

To run a specific graph, one must go to line 7 of the main.cpp program and change the bolded text that follows:

```
const char* filename = "g5.txt";
```

g1, g2, g3, etc to g10 are all part of the demo set.

To run the program, simply do:

```
./main          or          ./a.out
```

Data Structures Implemented

In this program, I use a simple 2D array to represent my graph as an adjacency matrix. It is outputted as such, as well as an adjacency list. All of the methods used are based around this array, as well as a bool array to be used as a stack.

Sufficient Conditions

To be able to draw a picture in one stroke, the graph must have one of the following:

1. Every vertex has an even degree
2. There are only 2 odd degree vertices

If the graph meets either of these two conditions, the graph can be drawn, and the program attempts to find the path.

Description of Algorithm

The algorithm starts at either vertex 1 if it satisfies condition 1, or the last odd degree vertex if it satisfies condition 2. It then traverses all adjacent vertices of the starting vertex. If there is only one adjacent vertex, it is outputted and removed. If there is more than one, it considers an adjacent vertex as long as the edge between the starting vertex and the adjacent vertex is not a bridge. To find if the edge is a bridge or not, it uses a DFS to return the number of vertices reachable from the starting vertex. It then removes the edge and finds the DFS of the vertex again, and if the number of reachable vertices are reduced, then the edge is indeed a bridge

$O((V+E)^2)$

Testing Evidence

```
compute.cse.tamu.edu - PuTTY
3 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0
4 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0
5 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0
6 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
7 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
8 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
11 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
13 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
14 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
15 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
16 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
17 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
18 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
19 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The graph is possible to be drawn in one stroke.
We should start at vertex 1.
1-2 2-6 6-1 6-4 4-8 8-3 3-7 7-8 8-5 5-6
[johnplunk]@compute ~/CSCE221/PA6> (04:19:55 04/30/19)
:: g++ -std=c++11 *.cpp
[johnplunk]@compute ~/CSCE221/PA6> (04:26:17 04/30/19)
```

```
compute.cse.tamu.edu - PuTTY
:: ./a.out
1 0 1 0 0 1
2 1 0 1 0 0
3 0 1 0 1 0
4 0 0 1 0 1
5 1 0 0 1 0
The graph is possible to be drawn in one stroke.
We should start at vertex 1.

[johnplunk]@compute ~/CSCE221/PA6> (03:21:00 04/30/19)
:: g++ -std=c++11 *.cpp
main.cpp:25:1: error: expected '}' at end of input
}
^
In file included from main.cpp:1:0:
graph.h: In member function 'void Graph::printPath(int)':
graph.h:47:54: error: 'isValidNextEdge' was not declared in this scope
    if (adjacencyMat[i][u] != 0 && isValidNextEdge(u,i) {
                                   ^
graph.h:47:56: error: expected ')' before '{' token
    if (adjacencyMat[i][u] != 0 && isValidNextEdge(u,i) {
                                   ^
graph.h:52:3: error: expected primary-expression before '}' token
}
```

```
compute.cse.tamu.edu - PuTTY
[johnplunk]@compute ~/CSCE221/PA6> (04:30:53 04/30/19)
:: ./a.out
1 0 1 1 1 0
2 1 0 1 0 0
3 1 1 0 1 0
4 1 0 1 0 1
5 0 0 0 1 0
The graph is not possible to be drawn in one stroke.

[johnplunk]@compute ~/CSCE221/PA6> (04:30:54 04/30/19)
:: g++ -std=c++11 *.cpp

[johnplunk]@compute ~/CSCE221/PA6> (04:31:21 04/30/19)
:: ./a.out
1 0 1 0 0 1
2 1 0 1 0 0
3 0 1 0 1 0
4 0 0 1 0 1
5 1 0 0 1 0
The graph is possible to be drawn in one stroke.
We should start at vertex 1.
1-2 2-3 3-4 4-5 5-1
[johnplunk]@compute ~/CSCE221/PA6> (04:31:22 04/30/19)
:: █
```