Class Graph

java.lang.Object
    Graph

```
public class Graph
extends java.lang.Object
```

Nested Class Summary

Nested Classes

Modifier and Type	Class and Description
`(package private) class`	**`Graph.MyLinkedList`** Makes a LinkedList always an int, so it can be placed in arrays

Constructor Summary

Constructors

Constructor and Description
`Graph``(int a)` Constructor

Method Summary

Methods

Modifier and Type	Method and Description
`void`	**`add`**`(int n, int k)` Method for adding a verticy
`java.lang.Boolean`	**`contains`**`(int i, int j)` Method for testing if an edge exists
`void`	**`DFS`**`()` Method which performs DFS search Will perform function to test DFS colors of verticies
`void`	**`DFSVisit`**`(int u)` Method which recursively calls itself to check every verticy
`boolean`	**`isAcyclic`**`()` Method to return cyclic variable First will perform DFS sort
`static void`	**`main`**`(java.lang.String[] argv)` Program to test functionality of the Graph class
`void`	**`print`**`()` Method for printing the graph Will return each element verticies on a line

`void`	`remove(int n, int k)` Method for removing a verticy
`Graph.MyLinkedList`	`top()` Method topologically sort the graph

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

Graph

```
public Graph(int a)
```

Constructor

Creates a new Graph

Method Detail

add

```
public void add(int n,
       int k)
```

Method for adding a verticy

Parameters:

`n` - link from

`k` - link to

remove

```
public void remove(int n,
          int k)
```

Method for removing a verticy

Parameters:

`n` - link from

`k` - link to

print

```
public void print()
```

Method for printing the graph Will return each element verticies on a line

contains

```
public java.lang.Boolean contains(int i,
                                  int j)
```

Method for testing if an edge exists

Parameters:

`i` - link from

`j` - link to

Returns:

boolean value of if g contains i->j

DFS

```
public void DFS()
```

Method which performs DFS search Will perform function to test DFS colors of verticies

DFSVisit

```
public void DFSVisit(int u)
```

Method which recursively calls itself to check every verticy

Parameters:

`n` - link from

`k` - link to

isAcyclic

```
public boolean isAcyclic()
```

Method to return cyclic variable First will perform DFS sort

Returns:

boolean value of cyclic

top

```
public Graph.MyLinkedList top()
```

Method topologically sort the graph

Returns:

MyLinkedList of topological sort

main

```
public static void main(java.lang.String[] argv)
```

Program to test functionality of the Graph class