Airline Route Simulator (Weighted Graph)

Each node represents a city and the edges connecting the nodes represent the cost of a ticket. Write a class named **Graph** that *implements* the **AirlineGraph** interface. The AirlineGraph interface is defined as:

```
interface AirlineGraph
{
    static final int SIZE = 10;

    static final String airportCode[] = { "BOS", "CHI", "DFW", "DEN",  "HNL",
                                          "IAH", "MIA", "JFK", "PHX", "SFO" };

    static final String city[] = {  "Boston, MA", "Chicago, IL", "Dallas-Ft Worth, TX",
                                    "Denver, CO", "Honolulu, HI", "Houston, TX",
                                    "Miami, FL", "New York, NY", "Phoenix, AX",
                                    "San Francisco, CA" };

    abstract String findRoute(int length, String start, String end);
}
```

In addition to the **public String findRoute(int length, String start, String end)** method required by the **AirlineGraph interface** your **Graph** class will need a **public String toString()** method among a number of other supporting methods. We will also be utilizing the Point class to store point A and point B destinations easier.

Run **AirlineRoutes**.java to test your code.

The Graph class will need to contain a stack and a structure to contain the graph.

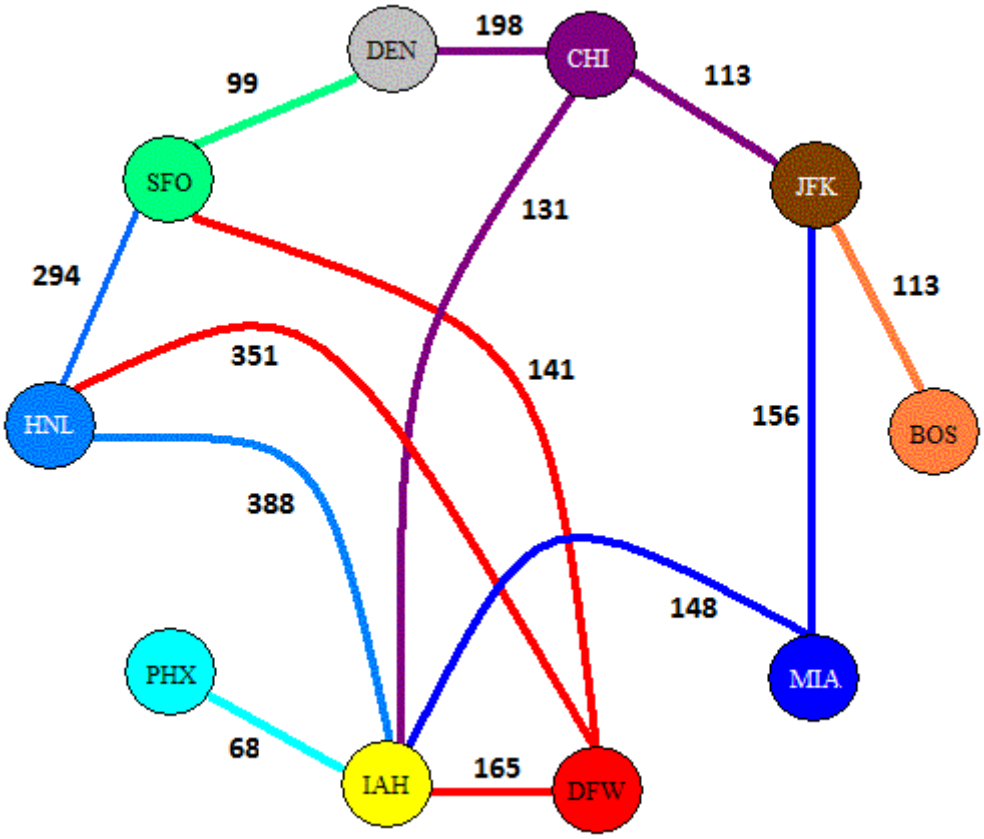
Field Summary

private int[][]	**graph** A two dimensional *adjacency matrix.*
private Stack<Integer>	**stack** A last in, first out data structure.

Constructor Summary

Graph()
Initializes a newly created Graph object. Initializes an empty stack and *adjacency matrix,* which must be a **10 X 10** grid. This method should also fill in the graph by setting the points contained in the data file "connections.dat" to be the designated value.

Method Summary	
public String	**toString()** Returns a String representation of this Graph object.
private int	**findAirportCode(String airportCode)** Returns the index position of the specified airportCode.
private boolean	**adjacent(Point edge)** Returns true if Point edge is connected, false otherwise.
public int[]	**shortestPath(String source)** Takes in an airport code and returns an array filled with the lowest cost to get from the source to every other city/airport. **Think about what we did by hand to get the path from end to start**
public String	**cheapestRoute (String start, String end)** Takes in 2 airport codes and returns a String containing all the cities visited from <*start*> to <*end*> in the order visited **followed by the total fare.** This method should prioritize the cost of the trip.
private boolean or int	**findPath(int length, Point p)** Returns true if a path of length <length> exists between the two points in Point/Edge <p> OR returns the cost to go from start to finish. If a path exists, the index values for each airport visited is pushed onto the stack. **boolean Algorithm** *(assuming p.x is the start and p.y is the end)***:** **if length equals 1** **if adjacent(p)** **push the ending city onto the stack** **return true** **else** **for every *node* in the graph** **if (*node* is adj. to p.x) && findPath(length – 1, Point(*node*, p.y))** **push the current city/*node* onto the stack** **return true** **return false** **int Algorithm:** **…**
public String	**findRoute(int length, String start, String end)** If a path with the specified length exists, return a String containing all the cities visited from <*start*> to <*end*> in the order visited **followed by the total fare.** Return *"There is no such connection!"* otherwise. This method should prioritize the length of the path rather than the cost. **Doesn't findPath(…) push cities onto a stack, interesting…**



Airline Graph

	BOS	CHI	DFW	DEN	HNL	IAH	MIA	JFK	PHX	SFO
BOS								113		
CHI				89		131		113		
DFW					388	165				123
DEN		198								194
HNL			351			388				352
IAH		98	165		407		128		99	
MIA						148		156		
JFK	113	113					148			
PHX						68				
SFO			141	99	294					

A Matrix representation of the Airline Graph

OUTPUT

```
      BOS  CHI  DFW  DEN  HNL  IAH  MIA  JFK  PHX  SFO
BOS    -    -    -    -    -    -    -   113   -    -

CHI    -    -    -   89    -  131    -   113   -    -

DFW    -    -    -    -  388  165    -    -    -  123

DEN    -  198    -    -    -    -    -    -    -  194

HNL    -    -  351    -    -  388    -    -    -  352

IAH    -   98  165    -  407    -  128    -   99    -

MIA    -    -    -    -    -  148    -  156    -    -

JFK  113  113    -    -    -    -  148    -    -    -

PHX    -    -    -    -    -   68    -    -    -    -

SFO    -    -  141   99  294    -    -    -    -    -

CheapestRoute Tests:
    Cheapest Route From SFO to JFK
    San Francisco, CA -> Denver, CO -> Chicago, IL -> New York, NY $410

    Cheapest Route From PHX to MIA
    Phoenix, AZ -> Houston, TX -> Miami, FL $196

    Cheapest Route From HNL to BOS
    Honolulu, HI -> Houston, TX -> Chicago, IL -> New York, NY -> Boston, MA $712

FindRoute Tests:
    SFO to JFK, 1 hop
    There is no such connection!

    PHX to MIA, 2 hops
    Phoenix, AZ -> Houston, TX -> Miami, FL  $196.00

    SFO to JFK, 4 hops
    San Francisco, CA -> Dallas-Ft Worth, TX -> Houston, TX -> Chicago, IL -> New
York, NY  $517.00

    HNL to BOS, 5 hops
    Honolulu, HI -> Dallas-Ft Worth, TX -> Houston, TX -> Chicago, IL -> New York,
NY -> Boston, MA  $840.00
```