CS 314 Final Review — Bipartite Graphs

Graphs

Write an instance method for the provided `Graph` class which determines whether a graph is bipartite. A graph is bipartite if the set of vertices can be partitioned into two disjoint sets such that no edges exist between nodes of the same subset. One way to determine if a graph is bipartite is to check if the graph can be colored using 2 colors. If each vertext can be colored with either one of two colors such that no two vertices of the same color are adjacent, then the graph is bipartite.

Assume we are only dealing with unweighted, undirected graphs.

Complete the following instance method.

```
// Determines whether or not a graph is bipartite.
// pre: vertices.size() > 0
// post: the structure of the graph is not altered by this operation
public boolean isBipartite() {
```

Use the following Graph implementation.

```
public class Graph {
  // The vertices in the graph.
  private Map<String, Vertex> vertices;

  // Sets scratch to 0 for all vertices
  private void clearAll(){ /* ... */ }

  private static class Vertex {
    private String name;
    private List<Edge> adjacent;
    private int scratch;
  }

  private static class Edge {
    private Vertex dest;
  }

}
```

You may create helper methods.
You may not create any new data structures.
Do not use any other Java classes or methods.

```
// Determines whether or not a graph is bipartite.
// pre: none
// post: returns true iff graph is bipartite
public boolean isBipartite() {
```