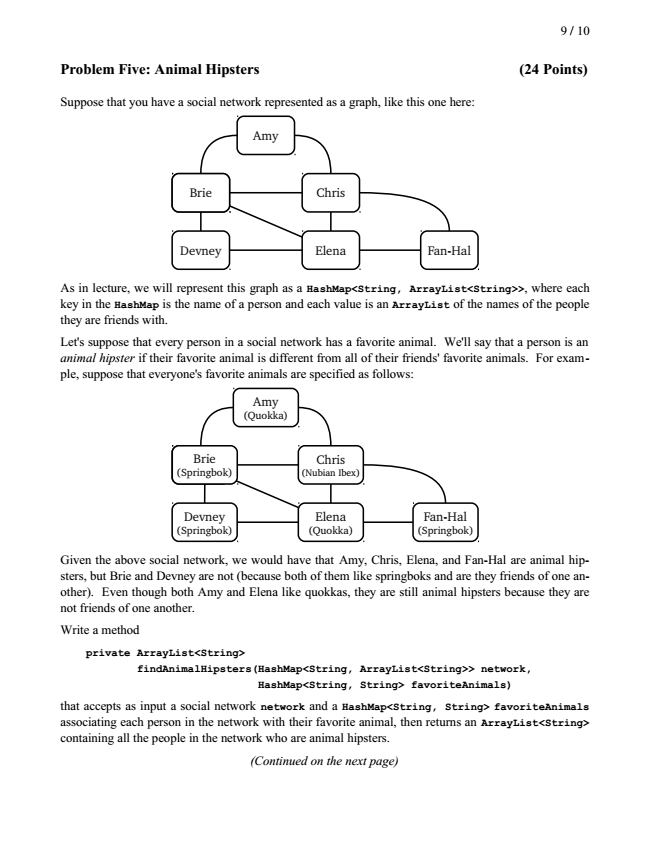
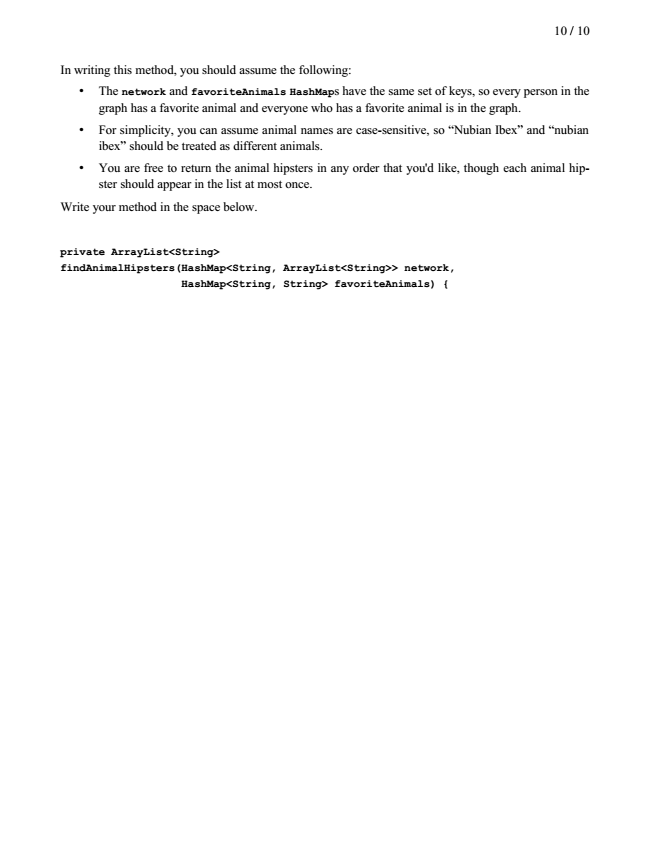
*This exercise is taken from Stanford’s CS106 midterm exam. Use the unit tests underneath the exercise to check your algorithm. Do not change the tests.*





// Copyright Wintriss Technical Schools 2014

import static org.junit.Assert.\*;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.HashMap;

import org.junit.Test;

public class AnimalHipsterTest

{

@Test

public void testAnimalHipster()

{

AnimalHipster animalHipster = new AnimalHipster();

HashMap<String, ArrayList<String>> network = new HashMap<String, ArrayList<String>>();

HashMap<String, String> favoriteAnimals = new HashMap<String, String>();

network.put("Amy", new ArrayList<String>(Arrays.asList(new String[] {"Brie", "Chris"})));

network.put("Brie", new ArrayList<String>(Arrays.asList(new String[] {"Amy", "Chris", "Elena", "Devney"})));

network.put("Chris", new ArrayList<String>(Arrays.asList(new String[] {"Amy", "Brie", "Elena", "Fan-Hal"})));

network.put("Devney", new ArrayList<String>(Arrays.asList(new String[] {"Brie", "Elena"})));

network.put("Elena", new ArrayList<String>(Arrays.asList(new String[] {"Brie", "Chris", "Devney", "Fan-Hal"})));

network.put("Fan-Hal", new ArrayList<String>(Arrays.asList(new String[] {"Chris", "Elena"})));

favoriteAnimals.put("Amy", "Quokka");

favoriteAnimals.put("Devney", "Springbok");

favoriteAnimals.put("Brie", "Springbok");

favoriteAnimals.put("Chris", "Nubian Ibex");

favoriteAnimals.put("Elena", "Quokka");

favoriteAnimals.put("Fan-Hal", "Springbok");

assertTrue(animalHipster.findAnimalHipsters(network, favoriteAnimals).containsAll(new ArrayList<String>(Arrays.asList(new String[] {"Amy", "Elena", "Chris", "Fan-Hal"}))));

}

@Test

public void testAnimalHipsterWithEmptyInput()

{

AnimalHipster animalHipster = new AnimalHipster();

HashMap<String, ArrayList<String>> network = new HashMap<String, ArrayList<String>>();

HashMap<String, String> favoriteAnimals = new HashMap<String, String>();

assertTrue(animalHipster.findAnimalHipsters(network, favoriteAnimals).isEmpty());

}

@Test

public void testAnimalHipsterWithEmptyNetwork()

{

AnimalHipster animalHipster = new AnimalHipster();

HashMap<String, ArrayList<String>> network = new HashMap<String, ArrayList<String>>();

HashMap<String, String> favoriteAnimals = new HashMap<String, String>();

network.put("Amy", new ArrayList<String>());

network.put("Brie", new ArrayList<String>());

network.put("Chris", new ArrayList<String>());

network.put("Devney", new ArrayList<String>());

network.put("Elena", new ArrayList<String>());

network.put("Fan-Hal", new ArrayList<String>());

favoriteAnimals.put("Amy", "Quokka");

favoriteAnimals.put("Devney", "Springbok");

favoriteAnimals.put("Brie", "Springbok");

favoriteAnimals.put("Chris", "Nubian Ibex");

favoriteAnimals.put("Elena", "Quokka");

favoriteAnimals.put("Fan-Hal", "Springbok");

assertEquals(new HashSet<>(Arrays.asList(new String[] {"Amy", "Brie", "Chris", "Devney", "Elena", "Fan-Hal"})), new HashSet<>(animalHipster.findAnimalHipsters(network, favoriteAnimals)));

}

}