Network Science
Homework Assignment #2
Dominic Farolino

> **Run NS3A on some connected network to find diameter.**

I modified NS3A.java to print the diameter of a given undirected connected network. When run on stringentNet.txt from the first assignment, a connected network, the output was a diameter of **6**.

When run on the input:
0 1
1 0 2
2 1
3 4
4 3

which represents the graph G = (V, E), V = {0, 1, 2, 3, 4}, E = {0⇒1, 1⇒0, 1⇒2, 2⇒1, 3⇒4, 4⇒3}, the diameter output is: **2**, which is correct.

> **Run NS3A on some network to find connected components.**

When run on stringentNet.txt as before, the program finds one connected component of size **4319** (whose diameter has been pointed out to be 6). When run on the custom graph above, the program finds two connected components, of size **3** and **2**, which is correct.

```
dfarolino@dfarolino:/Users/dfarolino/Desktop/Git/network-science/Assignment2  git:(master*) $ java NS3A undirectedGraph
Component 1 has 3 nodes.
Component 2 has 2 nodes.
Diameter of the graph: 2
[dfarolino@dfarolino:/Users/dfarolino/Desktop/Git/network-science/Assignment2  git:(master*) $ java NS3A ../Assignment1/stringentNet.txt
Component 1 has 4319 nodes.
Diameter of the graph: 6
```

> **Complete NS3B.java and run it on a directed network to find all strongly connected components.**

Output:
- When run on celegansOutNeighbors.txt
    - **51** strongly connected components
    - **Two** components of size > 1; sizes: **239**, and **2**
- When run on WikiVoteOutNeighbors.txt
    - **1740** strongly connected components
    - A **single** component of size > 1; size: **1300**

```
[dfarolino@dfarolino:/Users/dfarolino/Desktop/Git/network-science/Assignment2  git:(master*) $ java NS3B celegansOutNeighbors.txt
 0 239
 3 2
 Number of strongly connected components 51
[dfarolino@dfarolino:/Users/dfarolino/Desktop/Git/network-science/Assignment2  git:(master*) $ java NS3B WikiVoteOutNeighbors.txt
 0 1300
 Number of strongly connected components 1740
 dfarolino@dfarolino:/Users/dfarolino/Desktop/Git/network-science/Assignment2  git:(master*) $
```

> Complete NS3C.java and run it on a bipartite network to find all connected components.

After completing the NS3C file, I ran the program on the YgenotypesNeighbors bipartite graph and got output like the following:

```
[dfarolino@dfarolino:/Users/dfarolino/Desktop/Git
 Component 0 1233 1233
 Component 1 1 0
 Component 2 1 0
 Component 3 1 0
 Component 4 1 0
 Component 5 1 0
 Component 6 1 0
 Component 7 1 0
 Component 8 1 0
 Component 9 1 0
 Component 10 1 0
 Component 11 1 0
 Component 12 1 0
 Component 13 1 0
 Component 14 1 0
 Component 15 1 0
 Component 16 1 0
 Component 17 1 0
 Component 18 1 0
 Component 19 1 0
 Component 20 1 0
 Component 21 1 0
 Component 22 1 0
 Component 23 1 0
 Component 24 1 0
 Component 25 1 0
 Component 26 1 0
 Component 27 1 0
 Component 28 1 0
 Component 29 1 0
 Component 30 1 0
 Component 31 1 0
 Component 32 1 0
```

...this output continues all the way to the bottom. Please email me to request the full output if necessary.

I also ran the program on a custom bipartite graph below:

```
0 1
1 2
2 3
3 2
4 5
5 6
6 5 7
7 9
8 7
9 7
```

...and got the following output:

```
[dfarolino@dfarolino:/Users/dfarolino/Desktop/Git/network-science/Assignment2
Component 0 1 1
Component 1 2 1
Component 2 1 1
Component 3 4 2
Component 4 1 1
Component 5 1 1
Isolated item 0
Isolated item 4
Isolated item 8
```

The changes made to NS3C.java are as follows:

```java
void makeBackwardNet() {  // make inNeighbors from outNeighbors, need your code
  for (int i = 0; i < M; i++)
    inNeighbors.add(new HashSet<Integer>());  // all empty

  // your code to fill them
  for (int i = 0; i < M; i++) {
    for (int j: outNeighbors.get(i)) {
      inNeighbors.get(j).add(i);
    }
  }
}
```

```java
void expandB(int item){
  for (int j : inNeighbors.get(item)) if (componentsA[j] == -1){
    // You complete the section
    componentsA[j] = numberOfComponents;
    forward.add(j);
    expandA(j);
  }
}
```

** Please email to request the full files if necessary