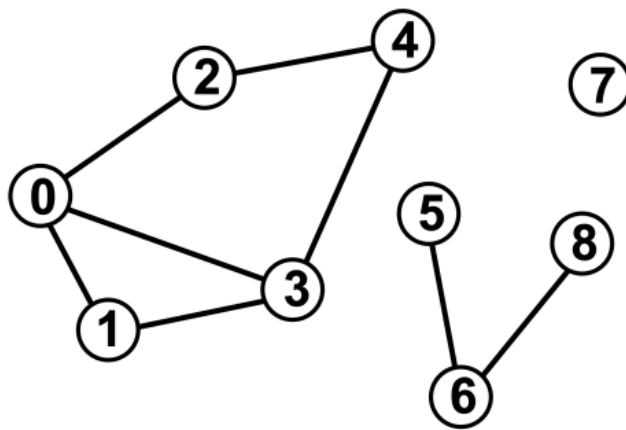


CSCI 232 Lab 6

Due Tuesday June 13th @ 11:59 PM. Please submit this assignment (.java files) to the appropriate dropbox on D2L

Background and Instructions

Start with the code linked below. The **UndirectedGraph** class is the same as the one we wrote in class on June 6th/7th, however the **UndirectedGraphDemo** class has been updated to reflect the following undirected graph:



You will write the body for one method in this lab. Complete the body of the **printConnectedComponents()** method so that it prints out each connected component as a set/list of vertices in that connected component. You may add any other methods or data structures you need to accomplish this task.

Valid output for this graph *could* look something like this:

```
{0, 1, 3, 4, 2}, {8, 5, 6}, {7}
```

Note: the order of the listed connected components does not matter and the order of the vertices in the connected components also does not matter.

Once you think you have your code working, try to add more edges/vertices to the graph to verify the correctness of your method.

Sample Output

When you run your program, it should print out the connected components of the graph. It does not need to look exactly like this, but the actual components should be correct (once again, the order does not matter).

```
[0, 1, 2, 3, 4]
[5, 6, 8]
[7]
```

Hints

- Consider using a HashSet to store the vertices of connected components
(HashSet<LinkedList<Integer>>)
- Consider using the **depthFirst()** method and **visited[]** array to figure out which nodes have been visited

Starting Code

- UndirectedGraph:
<https://www.cs.montana.edu/pearsall/classes/summer2023/232/labs/UndirectedGraph.java>
- UndirectedGraphDemo
<https://www.cs.montana.edu/pearsall/classes/summer2023/232/labs/UndirectedGraphDemo.java>

Grading

Grading will be done as follows:

- Program identifies all connected components – 10 points

NOTE: If your code does not compile, correctness cannot be verified, and you won't receive any points for your code. Turn in code that compiles!