CS 1501 Biconnected Components

Output from BiconnectedTrace.java using input file classGraph.txt. Each vertex has an indent level and each has a color assignment (but some vertices have the same color). The trace code shows how each vertex is assigned a DFS # (pre[v]) and a low[v] value that are initially equal. However, a vertex can reduce its low[v] value via:

1) back edges to "higher up" vertices in the tree, or
2) a child vertex having a smaller low[v] value.

These updates are indicated in the trace code with underlines.

If the child w of vertex v ends up with a low[w] value that is >= pre[v], it means that v is an articulation point. These are marked in **bold.**

Note that an edge from a child node "back" to a parent node in the tree is not considered a back edge and does not change the low[v] value for a vertex. To see this idea look at document bicon2.pdf. Note that vertex 8 has a low[v] value of 3 (in green), but it does have an edge to vertex 3, which has a DFS # of 2. It would seem from this that the low[v] for 8 should be 2, but since vertex 3 is the parent of vertex 8 in the tree, this edge is not considered. This does not affect the results of the execution.

```
fromWeb > java BiconnectedTrace classGraph.txt
9 11
0: 5 4 1
1: 5 0
2: 5 3
3: 8 5 2
4: 7 6 0
5: 3 2 1 0
6: 7 4
7: 6 4
8: 3

Visiting: 0
Initially: pre[v] = 0 and low[v] = 0
Recursing to child: 5
                        Visiting: 5
                        Initially: pre[v] = 1 and low[v] = 1
                        Recursing to child: 3
                Visiting: 3
                Initially: pre[v] = 2 and low[v] = 2
                Recursing to child: 8
                                        Visiting: 8
                                        Initially: pre[v] = 3 and low[v] = 3
                                        Final low[v] value for 8: 3
                Back to 3 after recursion
                Child 8 low[w]: 3
                Child 8 low[w] = 3 >= pre[v] = 2
                     3 is an articulation point
                Recursing to child: 2
            Visiting: 2
            Initially: pre[v] = 4 and low[v] = 4
            Back edge to 5 with DFS# 1
            1 < 4 so update low[v] to 1
            Final low[v] value for 2: 1
                Back to 3 after recursion
                Child 2 low[w]: 1
```

```
                    1 < 2 so update low[v] to 1
                    Final low[v] value for 3: 1
                              Back to 5 after recursion
                              Child 3 low[w]: 1
                              Child 3 low[w] = 1 >= pre[v] = 1
                                   5 is an articulation point
                              Back edge to 2 with DFS# 4
                              Recursing to child: 1
          Visiting: 1
          Initially: pre[v] = 5 and low[v] = 5
          Back edge to 0 with DFS# 0
          0 < 5 so update low[v] to 0
          Final low[v] value for 1: 0
                              Back to 5 after recursion
                              Child 1 low[w]: 0
                              0 < 1 so update low[v] to 0
                              Final low[v] value for 5: 0
Back to 0 after recursion
Child 5 low[w]: 0
Recursing to child: 4
                         Visiting: 4
                         Initially: pre[v] = 6 and low[v] = 6
                         Recursing to child: 7
                                        Visiting: 7
                                        Initially: pre[v] = 7 and low[v] = 7
                                        Recursing to child: 6
                                   Visiting: 6
                                   Initially: pre[v] = 8 and low[v] = 8
                                   Back edge to 4 with DFS# 6
                                   6 < 8 so update low[v] to 6
                                   Final low[v] value for 6: 6
                                        Back to 7 after recursion
                                        Child 6 low[w]: 6
                                        6 < 7 so update low[v] to 6
                                        Final low[v] value for 7: 6
                         Back to 4 after recursion
                         Child 7 low[w]: 6
                         Child 7 low[w] = 6 >= pre[v] = 6
                              4 is an articulation point
                         Back edge to 6 with DFS# 8
                         Final low[v] value for 4: 6
Back to 0 after recursion
Child 4 low[w]: 6
Back edge to 1 with DFS# 5
Root 0 has 2+ children
     0 is an articulation point
Final low[v] value for 0: 0

Articulation points
-------------------
0
3
4
5
```