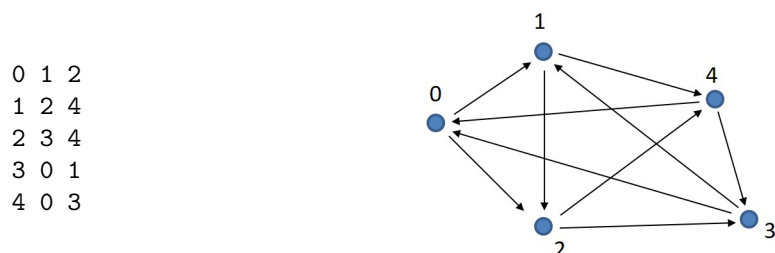


## CS 405: Algorithm Analysis II

### Homework 2: Graph Cycle

You are to write a Java program (Eclipse project) that will accept a directed graph description in adjacency list form and either (a) output the vertices, starting at vertex 0, of a cycle that traverses each **edge** exactly once, or (b) announce that the graph admits no such cycle. The input file has the form on the left below and corresponds to the graph on the right.



The first entry on each line is a vertex number. The remaining entries are the neighbors of that vertex. The first line of the example indicates that (0,1) and (0,2) are edges in the graph. This particular graph, which is available as g1.txt, does admit a cycle of the type described, namely



The required output is just the text, the cycle, not the picture. To accommodate larger graphs, you should format the output cycle, if you find one, with 20 vertices per line. In the example, you can check that each edge is traversed exactly once and the cycle begins and ends with zero.

Your algorithm should be  $\Theta(V + E)$ , which is the time required to read the input. Your Java program can use the ArrayList class as though it were a linked list in terms of performance. That is, you can add a node anywhere in the list in  $\Theta(1)$  time, **provided** you do not have to search for the insertion point. Assuming that you know the insertion point, that operation is simply swapping a few pointers, so we will assume that the ArrayList class avails itself of that efficiency. Otherwise, you could use a linked list, but that requires implementing all of the pointer swaps.

Please submit the following files in an email prior to the assignment deadline.

1. A copy of the output (.txt file) corresponding to the input on g2.txt.
2. A copy of your Eclipse project as a .zip file
3. A writeup (L<sup>A</sup>T<sub>E</sub>X-prepared .pdf file) that describes how you achieve your goal and why the complexity is  $\Theta(V + E)$ , linear in the graph size.