Please complete the following questions. For 1 and 2, please use Java. For 3, you can use any language you prefer. Compilable/runnable code is required, runnable code with tests that I can easily run to verify your submission is always preferred.

**Please publish to Github and email me the link or create a .tar.gz file.**

1. Imagine we have an interface "GNode" that looks like this:

   public interface GNode {

     public String getName();

    public GNode[] getChildren();

   }

   \*\* Observe that this GNode can be thought of as defining a graph.

   \*\* In implementing the functions below, you can assume that any

      graph defined by a GNode is acyclic (no cycles).

   \*\* Assume that when a GNode has no children, getChildren() returns

      a array of size 0, and \*not\* null.

   \*\* You can also assume that all children returned by getChildren()

      are not null.

   Implement a function with the following signature:

       public ArrayList walkGraph(GNode);

   which should return a ArrayList containing every GNode in the

   graph. Each node should appear in the ArrayList exactly once

   (i.e. no duplicates).

2. Implement a function with the following signature:

        public ArrayList paths(GNode node);

   which should return a ArrayList of ArrayLists, representing all

   possible paths through the graph starting at 'node'. The ArrayList

   returned can be thought of as a ArrayList of paths, where each path

   is represented as an ArrayList of GNodes.

   Example:

   Assume the following graph:

   A

     B

       E

       F

     C

       G

       H

       I

     D

       J

   paths(A) = ( (A B E) (A B F) (A C G) (A C H) (A C I) (A D J) )

3. Write a quick and dirty program (Shell, Python, Perl, Java, Lisp,

   C++, APL, or whatever) to produce a count of all the different

   "words" in a text file.  Use any definition of word that makes

   logical sense or makes your job easy.  The output might look like

   this:

     17 a

     14 the

      9 of

      9 in

      8 com

      7 you

      7 that

      7 energy

      6 to

      ...

   For this input file, the word "a" occured 17 times, "the" 14 times,

   etc.