Coding Test

Instructions:

1. Please complete the following questions.
2. You can use any coding language (preferably Python or TypeScript) of your choosing.
3. Code must be compilable/runnable such that we can easily run and verify the correct functionality. The choice of the tool to compile, test, and run the code is up to you.
4. You may publish and email a link to your results in Github or provide a `.tar.gz file` with your results by email.

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

```
interface GNode {
    getName(): string
    getChildren(): GNode[]
}
```

- Note that `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 a directed acyclic graph (DAG).
- Assume that when a `GNode` has no children, `getChildren()` returns a list of size 0.

1.a. Implement a function with the following signature:

```
walkGraph(node: GNode): GNode[]
```

which should return a list containing every `GNode` in the graph starting with the given `node`. Each node should appear in the list exactly once meaning there will be no duplicates.

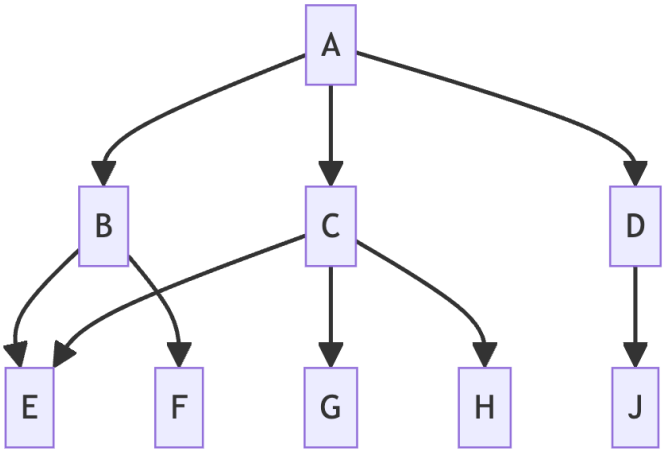
1.b. Implement a function with the following signature:

```
paths(node: GNode): GNode[][]
```

which should return a list of lists, representing all possible paths through the graph starting at the given `node` . The returned list can be thought of as a list of paths, where each path is represented as a list of `GNodes`.

Example

Assume the following graph:



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

2. Write a quick and dirty program

Write a script or program to produce a count of all the different "words" in a text file, sorted by count. Further, the words with the same count are sorted alphabetically (see for example the group of words that occurred seven times in the sample below). Use any definition of the word that makes logical sense or makes your job easy.

The output might look like this:

```
17 a
14 the
9 in
9 of
8 com
7 energy
7 that
7 you
6 to
...
```

For this input file, the word "a" occurred 17 times, "the" 14 times, etc.

3. Extra Credit (not required)

Write an HTTP server with REST API

You're creating a tool that helps people organize documents into different groups called "tags".

We can represent each document by a URI. Each document can have multiple tags. Tags are represented as text strings - you can enforce reasonable constraints (i.e. limit length, prevent white-spaces, etc.). Tags form a hierarchy, so each tag can have sub-tags. For example "animals" tag might have sub-tags like "mammals" and "birds" which can have sub-tags of their own. In other words the "is a sub-tag of" relation is transitive.

Implement an HTTP server that will expose the following API endpoint: `GET /taggedContent?tag=<tag>`

This returns a JSON array of documents (represented by URIs) associated with the given tag (and its sub-tags transitively).

The API is read-only so the "database" of documents and tags can be completely static. You can either hard-code it or preferably read it from a file.