Coding Question

7. Implement depth-first search in either C, C++, C#, Java, or Python. Given an undirected graph with n nodes and m edges, your code should run in $O(n + m)$ time. Remember to submit a makefile along with your code, just as with week 1's coding question.

   Input: the first line contains an integer t, indicating the number of instances that follows. For each instance, the first line contains an integer n, indicating the number of nodes in the graph. Each of the following n lines contains several space-separated strings, where the first string s represents the name of a node, and the following strings represent the names of nodes that are adjacent to node s. You can assume that the nodes are listed line-by-line in lexicographic order (0-9, then A-Z, then a-z), and the adjacent nodes of a node are listed in lexicographic order. For example, consider two consecutive lines of an instance:

   ```
   0, F
   B, C, a
   ```

   Note that 0 < B and C < a.

   Input constraints:

   - $1 \leq t \leq 1000$
   - $1 \leq n \leq 100$
   - Strings only contain alphanumeric characters
   - Strings are guaranteed to be the names of the nodes in the graph.

   Output: for each instance, print the names of nodes visited in depth-first traversal of the graph, *with ties between nodes visiting the first node in lexicographic order*. Start your traversal with the first node in lexicographic order. The names of nodes should be space-separated, and each line should be terminated by a newline.

   Sample:

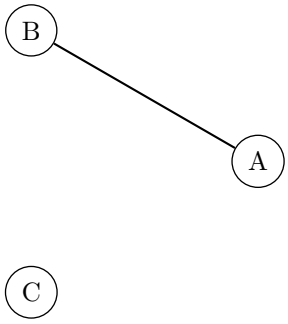
   Input:

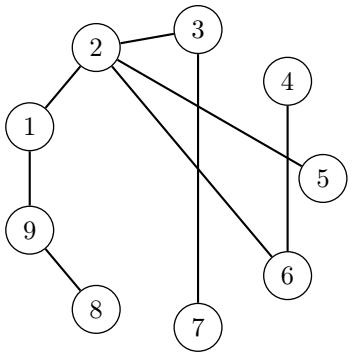
   ```
   2
   3
   A B
   B A
   C
   9
   1 2 9
   2 1 3 5 6
   3 2 7
   4 6
   5 2
   6 2 4
   7 3
   8 9
   9 1 8
   ```

   Output:

   ```
   A B C
   1 2 3 7 5 6 4 9 8
   ```

   The sample input has two instances. The first instance corresponds to the graph below on the left. The second instance corresponds to the graph below on the right.


B
A
C


3
2
4
1
5
9
6
8
7