Given a directed, acyclic graph of N nodes.  Find all possible paths from node 0 to node N-1, and return them in any order.

The graph is given as follows:  the nodes are 0, 1, ..., graph.length - 1.  graph[i] is a list of all nodes j for which the edge (i, j) exists.

**Example:**

**Input:** [[1,2], [3], [3], []]

**Output:** [[0,1,3],[0,2,3]]

**Explanation:** The graph looks like this:

0--->1

| |

v v

2--->3

There are two paths: 0 -> 1 -> 3 and 0 -> 2 -> 3.

**Note:**

* The number of nodes in the graph will be in the range [2, 15].
* You can print different paths in any order, but you should keep the order of nodes inside one path.