all_toposort.py

```python
from collections import defaultdict

def all_topological_sorts(AL, indegree, visited, stack, letters):
    global count

    flag = False

    for i in range(len(AL)):
        if indegree[i] == 0 and not visited[i]:
            for v in AL[i]:
                indegree[v] -= 1
            stack.append(i)
            visited[i] = True
            all_topological_sorts(AL, indegree, visited, stack, letters)
            visited[i] = False
            stack.pop()
            for v in AL[i]:
                indegree[v] += 1

            flag = True

    if not flag:
        count += 1
        ans = []
        for i in stack:
            ans.append(letters[i])
        if len(ans) != 0:
            print(' '.join(map(str, ans)))
        else:
            print("NO")

if __name__ == '__main__':
    t = int(input())
    for _ in range(t):
        input()
        letters = list(input().split())
        rules = list(input().split())
        mapper = {}
        for vertex in range(len(letters)):
            mapper[letters[vertex]] = vertex
        AL = [[] for _ in range(len(letters))]
        for edge in rules:
            v, u = edge.split('<')
            AL[mapper[v]].append(mapper[u])
        #print(letters, rules)
        indegree = [0] * len(letters)
        for u in AL:
            for v in u:
                indegree[v] += 1

        visited = [False] * len(letters)
        stack = []
        count = 0

        all_topological_sorts(AL, indegree, visited, stack, letters)
        if _ != t-1:
            print()


```