

< Back | Brief python solution using topological sort



luffy\_zc
 70
 Last Edit: October 21, 2018 9:21 AM
 2.2K VIEWS

The basic is to get topological sort of seqs nodes, if it is unique and equal to org, then true, else False

Following is how to implement in details:

in each step,

if we have more than one node whose incoming nodes count is zero then org is not unique, return False

At last we check if the topological sort contain all nodes in the in seqs and equal to org

```

from collections import defaultdict

class Solution(object):
    def sequenceReconstruction(self, org, seqs):
        """
        :type org: List[int]
        :type seqs: List[List[int]]
        :rtype: bool
        """
        adjacent = defaultdict(list)
        incoming_nodes = defaultdict(int)
        nodes = set()
        for arr in seqs:
            nodes |= set(arr)
            for i in xrange(len(arr)):
                if i == 0:
                    incoming_nodes[arr[i]] += 0
                if i < len(arr) - 1:
                    adjacent[arr[i]].append(arr[i + 1])
                    incoming_nodes[arr[i + 1]] += 1
        cur = [k for k in incoming_nodes if incoming_nodes[k] == 0]
        res = []
        while len(cur) == 1:
            cur_node = cur.pop()
            res.append(cur_node)
            for node in adjacent[cur_node]:
                incoming_nodes[node] -= 1
                if incoming_nodes[node] == 0:
                    cur.append(node)
            if len(cur) > 1:
                return False
        return len(res) == len(nodes) and res == org
    
```

[python](#)
[topological-sort](#)

Comments: 1

[Best](#)
[Most Votes](#)
[Newest to Oldest](#)
[Oldest to Newest](#)

Type comment here... (Markdown is supported)