CS3201 Algorithm Design
Semester 1/2018

Term Project Report
Problem: 1280
Timus Online Judge

Topological Sorting

Submitted to
Dr. Thitipong Tanprasert

Submitted by
5648042 Min Aung Dain
5918422 Sonam Choejur Tshering
1. Introduction 2
  1.1) Problem Detail
  1.2) Problem Description
  1.3) Input and Output

2. Code Overview 4
  2.1) Coding
  2.2) Code Explanation
  2.3) Output

3. Submission Result 6
  3.1) Timus Status
  3.2) Submission Detail
1. Introduction

1.1) Problem Details

Problem: 1280, Topological Sorting

Author: Vladimir N. Pinaev

Time limit: 1.0 second

Memory limit: 64MB

Difficulty level: 153

Problem source: 2003-2004 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk

1.2) Problem Description

Michael wants to win the world championship in programming and decided to study \( N \) subjects (for convenience we will number these subjects from 1 to \( N \)). Michael has worked out a study plan for this purpose. But it turned out that certain subjects may be studied only after others. So, Michael’s coach analyzed all subjects and prepared a list of \( M \) limitations in the form “\( s_i, u_i \)” (\( 1 \leq s_i, u_i \leq N; i = 1, 2, \ldots, M \)), which means that subject \( s_i \) must be studied before subject \( u_i \).

Your task is to verify if the order of subjects being studied is correct.

Remark. It may appear that it’s impossible to find the correct order of subjects within the given limitations. In this case any subject order worked out by Michael is incorrect.

Limitations

\( 1 \leq N \leq 1000; 0 \leq M \leq 100000. \)

1.3) Input and Output

Input

The first line contains two integers \( N \) and \( M \) (\( N \) is the number of the subjects, \( M \) is the number of the limitations). The next \( M \) lines contain pairs \( s_i, u_i \), which describe the order of subjects: subject \( s_i \) must be studied before \( u_i \). Further there is a sequence of \( N \) unique numbers ranging from 1 to \( N \) — the proposed study plan.
Output

Output a single word “YES” or “NO”. “YES” means that the proposed order is correct and has no contradictions with the given limitations. “NO” means that the order is incorrect.

Sample

<table>
<thead>
<tr>
<th>input</th>
<th>output</th>
</tr>
</thead>
</table>
| 5 6
1 3
1 4
3 5
5 2
4 2
1 2
1 3 4 5 2 | YES |
| 5 6
1 3
1 4
3 5
5 2
4 2
1 2
1 2 4 5 3 | NO |

2. Code Overview

2.1) Coding

```python
import sys
sys.setrecursionlimit(1000)

n, m = list(map(int, input().split()))
a = [0 for i in range(n+1)]
graph = [[] for _ in range(n+1)]

for i in range(m):
    s, u = list(map(int, input().split()))
    graph[s].append(u)
    a[u] = a[u] + 1

v = list(map(int, input().split()))
f = False

for i in range(n):
    if a[v[i]] == 0:
        a[v[i]] = -1
        for j in range(len(graph[v[i]])):
            a[graph[v[i]][j]] = a[graph[v[i]][j]] - 1
    elif a[v[i]] > 0:
        print('NO')
        f = True
        break
    if not f:
        print('YES')
```

2.2) Code explanation

For the test case sample 1

```
5 6
1 3
1 4
3 5
5 2
4 2
1 2
1 3 4 5 2
```
We declare 'n' and 'm' to accept the input of N subjects and M limitations respectively.

Initialize 'a' list with value 0 of size n+1.

Create 'graph' list that stores the values of u at index s.

Adds value of 1 in a at index u.

'v' list stores N sequence of numbers from the input(proposed study plan).

For loop that checks if value of a at index v[i] is equal to zero. If yes, value is changed to -1 to indicate value at that index in a is in the proposed study plan(v).

Then outer for loop of range n has if else statement that checks if the value at a[v[i]] == 0, if so it enters the inner for loop.

The inner for loop with range of length of graph at index v[i](value of v for outer loop i).

The value of graph at v[i][j] (graph[v[i]][j]) will be the index of ‘a’ list from which 1 is subtracted.

The above picture illustrates the working code and the highlights show the values being subtracted by 1.

Else, if the value at a[v[i]] > 0, print NO and turn the Boolean marker f = true and break.

2.3) Output

<table>
<thead>
<tr>
<th>input</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 6</td>
<td>YES</td>
</tr>
<tr>
<td>1 3</td>
<td></td>
</tr>
<tr>
<td>1 4</td>
<td></td>
</tr>
<tr>
<td>3 5</td>
<td></td>
</tr>
<tr>
<td>5 2</td>
<td></td>
</tr>
<tr>
<td>4 2</td>
<td></td>
</tr>
<tr>
<td>1 2</td>
<td></td>
</tr>
<tr>
<td>1 3 4 5 2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>input</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 6</td>
<td>NO</td>
</tr>
<tr>
<td>1 3</td>
<td></td>
</tr>
<tr>
<td>1 4</td>
<td></td>
</tr>
<tr>
<td>3 5</td>
<td></td>
</tr>
</tbody>
</table>
3. Submission Result

3.1) Timus status

Recent submissions

<table>
<thead>
<tr>
<th>ID</th>
<th>Date</th>
<th>Author</th>
<th>Problem</th>
<th>Language</th>
<th>Judgement result</th>
<th>Test #</th>
<th>Execution time</th>
<th>Memory used</th>
</tr>
</thead>
<tbody>
<tr>
<td>8133095</td>
<td>16:58:14 28 Nov 2018</td>
<td>Min Aung Dan</td>
<td>1293, Topological Sorting</td>
<td>Python 3.6</td>
<td>Accepted</td>
<td>0.992</td>
<td>2 568 KB</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Date</th>
<th>Author</th>
<th>Problem</th>
<th>Language</th>
<th>Judgement result</th>
<th>Test #</th>
<th>Execution time</th>
<th>Memory used</th>
</tr>
</thead>
<tbody>
<tr>
<td>8155491</td>
<td>20:38:10 28 Nov 2018</td>
<td>SonamCT</td>
<td>1292, Topological Sorting</td>
<td>Python 3.6</td>
<td>Accepted</td>
<td>0.992</td>
<td>2 572 KB</td>
<td></td>
</tr>
</tbody>
</table>

3.2) Submission Detail

Language: Python 3.6

Execution time: 0.592

Memory used: 2 568 KB