ROUTING PROBLEM

CHALLENGE DESCRIPTION:

A company has a network with H number of hosts. The network is customized according to the following rules:

1. Each host has 0 <= K <= 6 number of network interfaces (if there is no network interface, it means that a host is turned off).
2. Each interface is identified by an IP address and a subnet mask represented by CIDR notation (for example, 192.168.0.1 is an IP address, 255.255.255.0 is a NetMask, 192.168.0.1/24 is an IP address in CIDR).
3. Two hosts are on the same subnet if (IP_1 AND NetMask_1) = (IP_2 AND NetMask_2), where IP_i and NetMask_i are IP address and NetMask of the i-th computer, AND is a bitwise multiplication.
4. A package between two hosts in the same subnet is transmitted directly.
5. A package between two hosts that are located in different subnets can be transmitted through the third host in case their interfaces are connected to both source and destination subnets (for example, Host 1 has interfaces in the subnets "A" and "B", Host 2 has interfaces in the subnets "B and C", Host 3 has an interface in the subnet "C", Host 4 has an interface in the subnet "A", so the package path from 1 to 3 is [1, 2, 3], from 1 to 4 is [1, 4], and from 3 to 4 is [3, 2, 1, 4]).

Your task is to find all the shortest paths for the package between two specified hosts.

INPUT SAMPLE:

Your program should accept a path to a filename as its first argument. The first line of the file contains IDs of hosts and IP addresses that belong to each host (for example, {ID_0: ['IP address_1', 'IP address_K'], ID_1: [], ID_H: ['IP address_1', 'IP address_K']}), the other N lines are test cases. Each test case is a pair of host IDs (start_ID and end_ID) separated by a single space.

```
{0: ['179.177.220.225/23', '77.119.202.216/26', '171.251.60.138/25'], 1: ['147.105.101.114/22', '67.239.38.194/23',
'50.202.182.215/21'], 2: ['14.125.37.194/23', '128.3.0.158/23'], 3: ['77.119.202.193/26', '164.119.164.209/30'], 4:
['178.102.37.7/25', '117.184.65.161/21', '246.255.201.77/20', '128.160.69.237/30'], 5: ['67.239.38.85/23',
'128.3.1.67/23', '19.136.96.193/20', '108.243.129.176/24'], 6: ['103.131.199.252/25', '222.15.10.40/20'], 7:
['194.214.45.144/20', '22.53.223.185/26'], 8: ['147.105.102.57/22', '122.119.95.62/22', '0.111.125.205/30'], 9:
['0.111.125.206/30'], 10: ['117.184.70.222/21'], 11: ['178.102.37.31/25', '19.136.100.54/20', '96.43.12.49/28',
'214.240.75.20/23'], 12: ['171.251.60.130/25', '198.240.92.134/22', '242.183.157.69/27', '235.172.192.119/23'], 13:
['122.119.93.84/22'], 14: ['128.160.69.238/30', '22.53.223.131/26', '89.111.130.102/30'], 15: ['50.202.182.148/21',
'235.172.192.12/23'], 16: ['179.177.220.149/23', '103.131.199.146/25', '198.240.92.1/22'], 17: ['96.43.12.52/28'], 18:
['164.119.164.210/30', '108.243.129.104/24', '222.15.4.215/20', '89.111.130.101/30', '63.113.177.19/28'], 19:
['14.125.36.203/23', '246.255.204.239/20', '194.214.34.254/20', '214.240.74.123/23', '242.183.157.93/27',
'63.113.177.27/28']}
5 8
1 7
2 14
11 20
10 14
```

OUTPUT SAMPLE:

For each test case, print out the shortest path represented by a list of host IDs in the order from start_ID to end_ID. If there is no path, print out "No connection".

```
[5, 1, 8]
[1, 5, 2, 19, 7], [1, 5, 11, 19, 7], [1, 5, 18, 14, 7], [1, 5, 18, 19, 7], [1, 15, 12, 19, 7]
[2, 5, 18, 14], [2, 19, 4, 14], [2, 19, 7, 14], [2, 19, 18, 14]
No connection
[10, 4, 14]
```

CONSTRAINTS:

1. H is in a range from 150 to 300.
2. ID is in a range from 0 to H – 1.
3. K is in a range from 0 to 6.
4. N is in a range from 30 to 40.
5. IP address is in a range from 1.0.0.0 to 254.254.254.254.
6. NetMask is in a range from 20 to 30.

Sorting example:
Please use the natural sorting in case there is more than one path. E.g. for the lists [1,1,2,3], [0,2,1,2], [1,1,2,2], [0,1,1,2] the output is [0,1,1,2], [0,2,1,2], [1,1,2,2], [1,1,2,3]