HW6) Investors

Problem | Submissions | Leaderboard

Submitted a minute ago • Score: 35.00 Status: Accepted

✔ Test Case #0	✔ Test Case #1	✔ Test Case #2
✔ Test Case #3	✔ Test Case #4	✔ Test Case #5
✔ Test Case #6	✔ Test Case #7	

Submitted Code

Language: Python 3 Open in editor

```
def getNum(letter):
    return ord(letter) - ord("A")

def getLetter(num):
    return chr(num + ord("A"))

def intersection(L1, L2):
    L3 = []
    for e in L1:
        if e in L2:
            L3.append(e)
    return L3

def union(L1, L2):
    L3 = L1.copy()
    for e in L2:
        if e not in L1:
            L3.append(e)
    return L3

def AdjacencyMatrix(V, edges):
    graph = [[False]*V for _ in range(V) ]

    for edge in edges:
        X,Y = (edge)
        i, j = getNum(X),getNum(Y)

        graph[i][j] = True
        graph[j][i] = True
    return graph

def contacts(graph, investor):

```

```
    size = len(graph)
    family = [investor]
    stack = [investor]
    visited = [False]*size
    visited[investor] = True

    while stack:
        node = stack.pop()

        visited = [False]*size
        for i in family:
            visited[i] = True


        for i in range(size):
            if not visited[i]:
                visited[i] = True
                if graph[node][i]==True:
                    family.append(i)
                    stack.append(i)

    return family

def check_connection(graph, investor1, investor2):

    size = len(graph)

    contacts1 = contacts(graph, investor1)
    contacts2 = contacts(graph, investor2)

    inter = intersection(contacts1, contacts2)

    totalcontacts = union(contacts1, contacts2)
    totalcontacts.sort()
    totalcontactsletters = [getLetter(c) for c in totalcontacts]
    word = ""
    for c in totalcontactsletters:
        word = word + c

    if not inter:

        graph[investor1][investor2] = True
        graph[investor2][investor1] = True

        print("T", word)

    else:
        print("F", word)



#input treatment
##graph definition

graph_numbers = input()
graph_numbers = tuple(map(int,graph_numbers.split()))
V, E = graph_numbers

edges = []
for i in range(E):
    edge = input()
    edge = tuple(map(str,edge.split()))
    edges.append(edge)

graph = AdjacencyMatrix(V, edges)

```

```
##queries
num_queries = int(input())
for i in range(num_queries):
    q = input()
    q = tuple(map(str,q.split()))
    investor1, investor2 = q

    check_connection(graph, getNum(investor1), getNum(investor2))
```

Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy |