## **Ví dụ**

7

Isenbaev Oparin Toropov

Ayzenshteyn Oparin Samsonov

Ayzenshteyn Chevdar Samsonov

Fominykh Isenbaev Oparin

Dublennykh Fominykh Ivankov

Burmistrov Dublennykh Kurpilyanskiy

Cormen Leiserson Rivest

Ta có:

* Oparin, Toropov, Fominykh là bạn chung team với Isenbaev.

→ Có Isenbaev’s number = 1

* Ayzenshteyn, Samsonov, Dublennykh, Ivankov là bạn chung team với 1 trong các bạn Oparin, Toropov, Fominykh.

→ Có Isenbaev’s number = 2

* Chevdar, Burmistrov, Kurpilanskiy là bạn chung team với 1 trong các bạn Ayzenshteyn, Samsonov, Dublennykh, Ivankov.

→ Có Isenbaev’s number = 3

* Cormen, Leiserson, Rivest chưa từng là bạn chung team với bạn nào từng chung team với Isenbaev.

→ Có Isenbaev’s number = undefined

**Output:**

Ayzenshteyn 2

Burmistrov 3

Chevdar 3

Cormen undefined

Dublennykh 2

Fominykh 1

Isenbaev 0

Ivankov 2

Kurpilyanskiy 3

Leiserson undefined

Oparin 1

Rivest undefined

Samsonov 2

Toropov 1

## **Hướng giải:**

* Xem mỗi bạn là 1 đỉnh của đồ thị. 2 đỉnh sẽ có cạnh nối với nhau nếu 2 bạn đã từng chung team với nhau.
* Khi đó, con số Isenbaev number của từng bạn sẽ là khoảng cách giữa đỉnh tương ứng với bạn đó và đỉnh tương ứng với Isenbaev. → Cách cạnh không có trọng số nên dùng thuật BFS.
* Dùng CTDL Binary Search Tree để tạo từng cặp key là tên của các bạn, value là id của đỉnh tương ứng.

## **Giải thuật:**

B1: Đọc dữ liệu.

B2: Với mỗi bạn, tạo 1 id tương ứng và cho vào BST.

B3: Bắt đầu duyệt BFS từ đỉnh của bạn Isenbaev.

B4: Duyệt từng cặp (key, value) tăng dần theo key:

Nếu visited[value] == False:

Xuất key và “undefined”.

Ngược lại, xuất key và dist[value].

## **Mã giả:**

BFS(start):

q = Queue()

dist[start] = 0

q.add(start)

while q not empty:

u = q.top()

q.pop()

for v in graph[u]:

if visited[v] == False:

visited[v] = True

dist[v] = dist[u] + 1

main():

read(N)

id = 0

graph = [] // Danh sách đỉnh kề

map = Map() //BST với key là tên và value là id của đỉnh tương ứng

for i = 0 to N - 1:

read(name1, name2, name3)

if (name1 not in map):

id += 1

map.insert(name1, id)

id1 = id

if (name2 not in map):

id += 1

map.insert(name2, id)

id2 = id

if (name3 not in map):

id += 1

map.insert(name3, id)

id3 = id

graph[id1].append(id2)

graph[id1].append(id3)

graph[id2].append(id1)

graph[id2].append(id3)

graph[id3].append(id1)

graph[id3].append(id2)

MAX = 305

dist = [-1] \* MAX

visited = [False] \* MAX

BFS(map[“Isenbaev”]) // O(N + M)

for (key, value) in map:

if (visited[value] == False):

print(key, “ undefined”)

else:

print(key, dist[value])

Độ phức tạp: O(NlogN + N + M)

N : số đỉnh

M : số cạnh