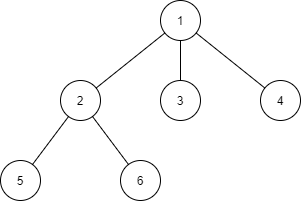
## Ví dụ

6 ← số thành phố

2 1

1 3

1 4

5 2

2 6

4 ← số cô gái

5

6

3

4

* Các cô gái ở gần Bishu nhất là cô gái ở nước 3 và 4.

⇒ Chọn cô gái ở nước có id nhỏ hơn ⇒ Output: 3

## Hướng giải

Cách 1:

* Với mỗi đất nước của các cô gái sống, dùng thuật DFS đến thành phố 1, tính khoảng cách và lấy giá trị nhỏ nhất.

⇒ Độ phức tạp: O(Q \* (V + E))

Cách 2:

* Tính khoảng cách nhỏ nhất từ đất nước 1 đến các nước còn lại.

⇒ Dùng thuật BFS hoặc DFS.

=> Độ phức tạp: O(V + E)

**(Lưu ý: DFS chỉ được dùng để tìm khoảng cách nhỏ nhất giữa 2 đỉnh ở trên cây)**

## 

## Giải thuật

B1: Đọc và khởi tạo danh sách đỉnh kề.

B2: Bắt đầu duyệt DFS từ đỉnh 1, dùng mảng dist để lưu khoảng cách nhỏ nhất từ đỉnh 1 đến các đỉnh còn lại.

B3: Duyệt qua các đất nước mà các cô gái sống, lấy id của nước có khoảng cách nhỏ nhất và id nhỏ nhất.

B4: Xuất kết quả.

## 

## Mã giả

dfs(start):

visited = [False] \* V

stack = Stack()

dist[start] = 0

stack.push(start)

while (!stack.empty()):

u = stack.pop()

for v in graph[u]:

if !visited[v]:

dist[v] = dist[u] + 1

visited = true

stack.push(v)

main():

read(V)

graph = []

for i = 0 to V - 1:

read(u, v)

graph[u].append(v)

graph[v].append(u)

dist = [-1] \* V

dfs(1)

read(Q)

minDist = 1005

id = 0

for i = 0 to Q:

read(u)

if (minDist > dist[u] or (minDist == dist[u] and id > u):

midDist = dist[u]

id = u

print(id)

Độ phức tạp: O(V + E) = O (V + (V - 1)) = O (V)