

SARTAJ AND PATTERN

We can solve this problem by first thinking about 1-D solution, that is if a longest graph is given, then the node which will minimize the height will be mid node if total node count is odd or mid two node if total node count is even. This solution can be reached by following approach - Start two pointers from both end of the path and move one step each time, until pointers meet or one step away, at the end pointers will be at those nodes which will minimize the height because we have divided the nodes evenly so height will be minimum.

Same approach can be applied to a general tree also. Start pointers from all leaf nodes and move one step inside each time, keep combining pointers which overlap while moving, at the end only one pointer will remain on some vertex or two pointers will remain at one distance away. Those node represent the root of the vertex which will minimize the height of the tree. So we can have only one root or at max two root for minimum height depending on tree structure as explained above. For implementation we will not use actual pointers instead we'll follow BFS like approach, In starting all leaf node are pushed into the queue, then they are removed from tree, next new leaf node are pushed in queue, this procedure keeps on going until we have only 1 or 2 node in our tree.

Then we can apply DFS approach for finding the minimum height of the tree considering the above found nodes as root.

If this height is smaller than or equal to '**k**' then return '**YES**' else return '**NO**'.

Solution

"""

Created on Wed Aug 28 22:30:01 2019

@author: Ravi

"""

```
from collections import deque
for _ in range(int(input())):
    n,k=map(int,input().split())
    a=[[ ] for i in range(n)]
```

```

deg=[0]*n
for i in range(n-1):
    u,v=map(int,input().split())
    a[v-1].append(u-1)
    a[u-1].append(v-1)
    deg[u-1]+=1
    deg[v-1]+=1
b=deque()
vis=[0]*n
for i in range(n):
    if deg[i]==1:
        b.append(i)
while len(b)>2:
    for i in range(len(b)):
        x=b.popleft()
        n-=1
        for j in a[x]:
            deg[j]-=1
            if deg[j]==1:
                b.append(j)
st=[(b[0],1)]
he=0
vis[b[0]]=1
while st:
    x,h=st.pop()
    he=max(h,he)
    for i in a[x]:
        if vis[i]==0:
            vis[i]=1
            st.append((i,h+1))
if he<=k:
    print('YES')
else:
    print('NO')

```