

树上随机游走

以 r 为根构造一个有根树，假设结点 i 的深度为 $depth[i]$ ，并规定 $depth[r] = 0$ 。由于 langgod 无论如何移动，移动前后的结点深度只会 $+1$ 或者 -1 （只能向父亲结点或儿子结点移动）。因此，langgod 所在的最终结点 u 的深度的奇偶性和 t 的奇偶性一致。又因为 langgod 会移动 t 步，因此 $depth[u] \leq t$ 。综上，langgod 可能在所有满足 $depth[u] \leq t$ 且 $depth[u] \equiv t \pmod{2}$ 的结点上。

通过 DFS 记录每个结点的深度，然后判断每个结点的深度是否满足上述情况即可。本题也可以通过 BFS 记录各节点与 r 结点的距离完成。

```
#include<bits/stdc++.h>
#define Nmax 200010
#define int long long
#define IOS ios::sync_with_stdio(false);cin.tie(nullptr);cout.tie(nullptr);
using namespace std;
int n,r,t,depth[Nmax];
vector<int> edge[Nmax];
void dfs(int x,int f)
{
    depth[x]=f==0?0:depth[f]+1;
    for(int i:edge[x])
        if(i!=f)
            dfs(i,x);
}
signed main()
{
    IOS;
    cin>>n>>r>>t;
    for(int i=1;i<=n;i++)
    {
        int u,v;
        cin>>u>>v;
        edge[u].push_back(v);
        edge[v].push_back(u);
    }
    dfs(r,0);
    queue<int> q;
    for(int i=1;i<=n;i++)
        if(depth[i]<=t&&depth[i]%2==t%2)
            q.push(i);
    cout<<q.size()<<endl;
    while(!q.empty())
    {
        cout<<q.front()<<" ";
        q.pop();
    }
}
```