```cpp
#include <bits/stdc++.h>
using namespace std;

int n,m;
map<string,int> mapa;
vector<int> parent,r;
vector<int> filhos;
int findSet(int i) {
        return (parent[i] == -1) ? i : parent[i] = findSet(parent[i]);
}
bool isSameSet(int i, int j) {
        return findSet(i) == findSet(j);
}
void unionF(int i,int j){
    if (!isSameSet(i, j)) {
        int x = findSet(i), y = findSet(j);
        if (r[x] > r[y]) {
            filhos[findSet(x)] += filhos[findSet(y)];
            parent[y] = x;
        } else {
            filhos[findSet(y)] += filhos[findSet(x)];
            parent[x] = y;
            if (r[x] == r[y])
                r[y]++;
        }
    }
}
main(){
    int i,j,k;
    string a,b;

    cin >> n;

    for(i=0;i<n;i++){
        cin >> m;
        for(j=0;j<m;j++){
            cin >> a >> b;
            if(!mapa.count(a)){
                int aux = mapa.size();
                mapa[a] = aux;
                parent.push back(-1);
                filhos.push back(1);
                r.push back(0);
            }
            if(!mapa.count(b)){
                int aux = mapa.size();
                mapa[b] = aux;
                parent.push back(-1);
                filhos.push back(1);
                r.push back(0);
            }
            unionF(mapa[a],mapa[b]);
            cout << filhos[findSet(mapa[a])] << endl;
        }
        mapa.clear();
        parent.clear();
        filhos.clear();
        r.clear();
    }
}

```