树哈希

Tree Hashing

有根树

$$f_x = 1 + \sum_{y \in \text{son}_x} f_y \times \text{prime}(\text{size}_y)$$

其中，$\text{prime}(i)$ 表示第 i 个质数。

```
ULL Hash[N];
int sz[N];
void getHash(int x, int f){ // tree rooted at x
    sz[x] = 1; Hash[x] = 1;
    for(auto &to : edge[x]){
        if(to == f) continue;
        getHash(to, x);
        sz[x] += sz[to];
        Hash[x] += pList[sz[to]] * Hash[to];
    }
}
```

无根树

鉴于树是无根的，我们选取**树的重心**作为根节点开始哈希。注意一棵树的重心最多有两个且相邻，可以分别做哈希，把值存在一起。

```
int size[N], mxson[N];
void getCentroid(int x, int f, int &rt){ // rt is the id of centroid
    mxson[x] = 0, size[x] = 1;
    for(auto &to : edge[x]){
        if(to == f) continue;
        getCentroid(to, x, rt);
        size[x] += size[to];
        mxson[x] = max(mxson[x], size[to]);
    }
    mxson[x] = max(mxson[x], n - size[x]);
    if(rt == 0 || mxson[rt] > mxson[x]) rt = x;
}

ULL Hash[N];
int sz[N]; // pay attention that sz[] is different from size[]
void getHash(int x, int f){ // tree rooted at x
    sz[x] = 1; Hash[x] = 1;
    for(auto &to : edge[x]){
        if(to == f) continue;
        getHash(to, x);
        sz[x] += sz[to];
        Hash[x] += pList[sz[to]] * Hash[to];
    }
}
```