

# 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。

```
1  ULL Hash[N];
2  int sz[N];
3  void getHash(int x, int f){ // tree rooted at x
4      sz[x] = 1; Hash[x] = 1;
5      for(auto &to : edge[x]){
6          if(to == f) continue;
7          getHash(to, x);
8          sz[x] += sz[to];
9          Hash[x] += pList[sz[to]] * Hash[to];
10     }
11 }
```

### 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。

```
1  int size[N], mxson[N];
2  void getCentroid(int x, int f, int &rt){ // rt is the id of centroid
3      mxson[x] = 0, size[x] = 1;
4      for(auto &to : edge[x]){
5          if(to == f) continue;
6          getCentroid(to, x, rt);
7          size[x] += size[to];
8          mxson[x] = max(mxson[x], size[to]);
9      }
10     mxson[x] = max(mxson[x], n - size[x]);
11     if(rt == 0 || mxson[rt] > mxson[x]) rt = x;
12 }
13
14 ULL Hash[N];
15 int sz[N]; // pay attention that sz[] is different from size[]
16 void getHash(int x, int f){ // tree rooted at x
17     sz[x] = 1; Hash[x] = 1;
18     for(auto &to : edge[x]){
19         if(to == f) continue;
20         getHash(to, x);
21         sz[x] += sz[to];
22         Hash[x] += pList[sz[to]] * Hash[to];
23     }
24 }
```

