

```
1 //日期: 2018/ 时间:
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 const int N = 100005;
6 //Union Find Set
7 int father[N];           //father[x]中存放元素x的父亲节点
8 //初始化
9 void initial(){
10     for(int i=1;i<=N;i++){
11         father[i]=i;      //一开始, 每一个元素都是独立的一个集合。
12     }
13 }
14
15 //寻找元素x所在集合的根节点
16 int findFather(int x){
17     while(x!=father[x])
18         x=father[x];
19     return x;
20 }
21
22 //合并两个集合, 先判断两个元素是否同属于同一个集合, 只有当两个元素属于不同集合才合 并。
23 //合并的过程一般是把其中一个集合的根节点的父亲指向另一个集合的根节点
24 void Union(int a,int b){
25     int FaA = findFather(a);
26     int FaB = findFather(b);
27     if(FaA != FaB){
28         FaA = father[FaB];    //合并
29     }
30 }
31
32 //查找时压缩路径
33 int findFather1(int x){
34     int a = x;
35     while(x != father[x]){
36         x = father[x];        //寻找根节点
37     }//x代表根节点。下面把路径上的所有节点的father都改成根节点
38
39     //从a开始。把a想象成一个指针。指针上移
40     while(a != father[a]){
41         int z = a;
42         a = father[a];
43         father[z] = x;
44     }
45     return x;
46 }
47
48 int main(){
49     .
50
51     return 0;
52 }
53
54
```