

# 并查集UFS

---

## 1.所有元素parent初始化为-1

```
int father[100100];
fill(father, father + 100100, -1);
```

## 2.查找操作 ( 加入路径压缩优化 )

```
int findFather(int x)
{
    int a = x;
    while(father[x] >= 0) x = father[x]; //向上一直回溯到集合根结点
    while(father[a] >= 0)
    {
        int z = a;
        a = father[a];
        father[z] = x; //路径上的每个点都指向根结点x
    }
    return x;
}
```

## 3.合并操作 ( 加入按秩归并优化 )

```
void Union(int a, int b)
{
    int faA = findFather(a);
    int faB = findFather(b);
    if(faA != faB)
        if(father[faA] < father[faB]) //A是大树
        {
            father[faA] += father[faB];
            father[faB] = faA; //把B挂到A上
        }
        else
        {
            father[faB] += father[faA];
            father[faA] = faB; //B是大树，把A挂到B上
        }
}
```