Disjoint Set（并查集）

- S包含k个互斥集合S1--Sk，动态的，联通分量
- 每个集合有一个代表元素

操作

> 并查集：合并，查找
> 不支持remove，split分拆集合

Makeset:

- 新建一个只包含x的集合，并添加到S中

Union（x，y）

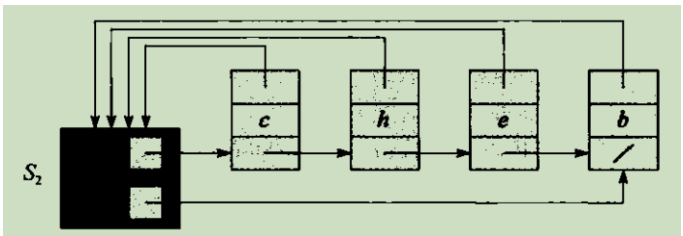
- 把x，y所在集合Sx和Sy取并集，替换掉Sx和Sy

> 每次操作后，减少一个集合
> 实际操作为把Sy加到Sx里面，代替删除

Find（x）

- 返回指向x所属集合代表元素的指针

实现方式一：链表



- 每个集合一个链表
- 链表每个元素包含集合成员：指向下一个的指针、指回集合对象的指针
- 第一个元素作为代表元素

Find（x）

- O（1）

- x是集合成员，包含两个指针和一个值，所以可以直接返回
- 沿着x对象的返回指针返回到集合对象，再返回head指向对象的成员

Union（x，y）

☐ 法一：如不区分长短 O（nn）

```
Makeset(x0)
for (i = 1 to n)//为了证明比较松的上界
    Makeset(xi)
    Union(xi,x0)
```

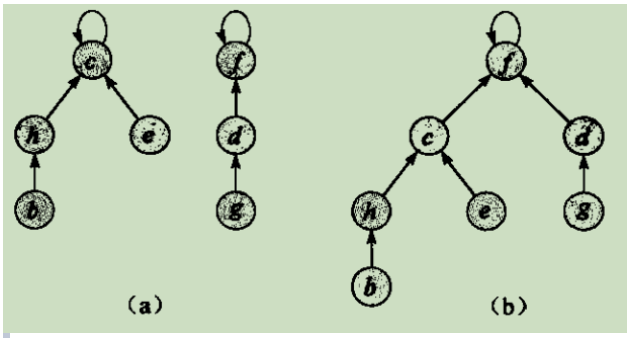
☐ 法二：所以把短的加在长的后面

- 时间复杂度和头指针改变次数严格相关

> 新加入的头指针要变化，旧的尾的有了下一个，原来的tail也要变化
> 每次union，短集合的规模至少翻倍，至多lgn次

- n次union一共O（nlgn）（非常松）
- 包含n个makeset的m个操作的序列，O（m+nlgn）

实现方式二：不相交集合森林



> 有根树

- 根是代表元素

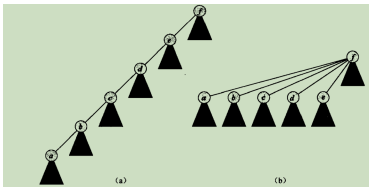
Find（x）

> 慢

- 沿着父指针找到根

优化：路径压缩：

- 找到根之后再遍历一次，把经过的所有结点的父节点指向根



Union（x，y）

> 快
>
> 只需要改一个结点的父指针

- 把x的根结点指向y的根

优化：按秩合并

- 较小的树挂在较大的上

> 除非高度一样，否则不改变高树的高度

- 结点秩至多lgn

> 可以归纳证明，按秩合并union保持此性质

cost分析

- makeset可先删掉
- 可把union都变成针对根节点的

Cost[**Union**(x, y)] = Cost[**Find**(x)] + Cost[**Find**(y)] + $O(1)$.

- 更改find

Partialfind(x,y)

> y是x祖先，以y为根路径压缩
>
> 所有find都可以替换成partialfind
>
> 目的，可以把union都提前

- partialfind与union有交换性