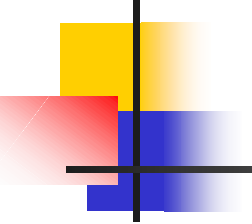


ACM程序设计

杭州电子科技大学 刘春英

acm@hdu.edu.cn





这一周，

你 **AC** 了吗？



每周一星（5）：

06092709朱卫江



第六讲

并查集

(Disjoint Set)



导引问题

在某个城市里住着 n 个人，任何两个认识的人不是朋友就是敌人，而且满足：

- 我朋友的朋友是我的朋友；
- 我敌人的敌人是我的朋友；

已知关于 n 个人的 m 条信息（即某2个人是朋友或者敌人），假设所有是朋友的人一定属于同一个团伙，请计算该城市最多有多少团伙？

如何实现？





什么是并查集？

英文：Disjoint Set，即“不相交集合”

将编号分别为 $1 \dots N$ 的 N 个对象划分为不相交集合，
在每个集合中，选择其中某个元素代表所在集合。

常见两种操作：

- 合**并**两个集合
- **查**找某元素属于哪个集合

所以，也称为“**并查集**”



实现方法（1）

- 用编号最小的元素标记所在集合；
- 定义一个数组 **set[1..n]**，其中**set[i]** 表示元素*i* 所在的集合；

i	1	2	3	4	5	6	7	8	9	10
Set(i)	1	2	1	4	2	6	1	6	2	2

不相交集合： {1,3,7}, {4}, {2,5,9,10}, {6,8}



方法(1)——效率分析

```
find1(x)
{
    return set[x];
}
```

$\Theta(1)$

```
Merge1(a,b)
{
    i = min(a,b);
    j = max(a,b);
    for (k=1; k<=N; k++) {
        if (set[k] == j)
            set[k] = i;
    }
}
```

$\Theta(N)$



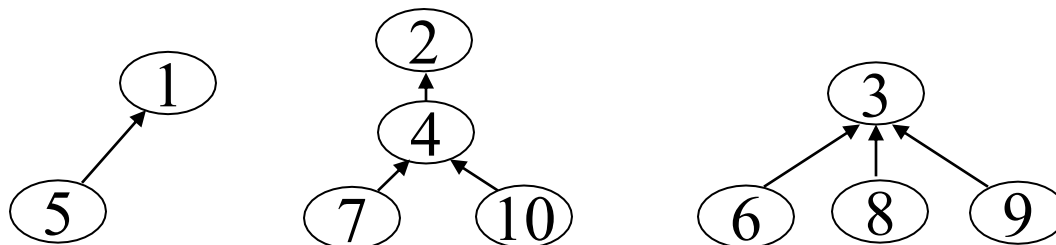
有待改进？

- 对于“合并操作”，必须搜索全部元素！
- 树结构如何？

实现方法 (2)

- 每个集合用一棵“有根树”表示
 - 定义数组 $\text{set}[1..n]$
 - $\text{set}[i] = i$, 则 i 表示本集合, 并是集合对应树的根
 - $\text{set}[i] = j, j \neq i$, 则 j 是 i 的父节点.

i	1	2	3	4	5	6	7	8	9	10
Set(i)	1	2	3	2	1	3	4	3	3	4





方法(2)——效率分析

```
find2(x)
{
    r = x;
    while (set[r] != r)
        r = set[r];
    return r;
}
```

最坏情况 $\Theta(N)$
一般情况是
...?

```
merge2(a, b)
{
    if (a < b)
        set[b] = a;
    else
        set[a] = b;
}
```

$\Theta(1)$



困惑~~~

- 性能有本质改进?
- 如何避免最坏情况?



避免最坏情况

- **方法**：将深度小的树合并到深度大的树
- **实现**：假设两棵树的深度分别为 h_1 和 h_2 ，则合并后的树的高度 h 是：
 - $\max(h_1, h_2)$, if $h_1 \neq h_2$.
 - $h_1 + 1$, if $h_1 = h_2$.
- **效果**：任意顺序的合并操作以后，包含 k 个节点的树的最大高度不超过 $\lceil \lg k \rceil$



优化后算法及效率

```
find2(x)
{
    r = x;
    while (set[r] != r)
        r = set[r];
    return r;
}
```

最坏情况 $\Theta(\log N)$

```
merge3(a,b)
{ if (height(a) == height(b)) {
    height(a) = height(a) + 1;
    set[b] = a;
} else if (height(a) < height(b))
    set[a] = b;
else
    set[b] = a; }
```

$\Theta(1)$



进一步优化——路径压缩

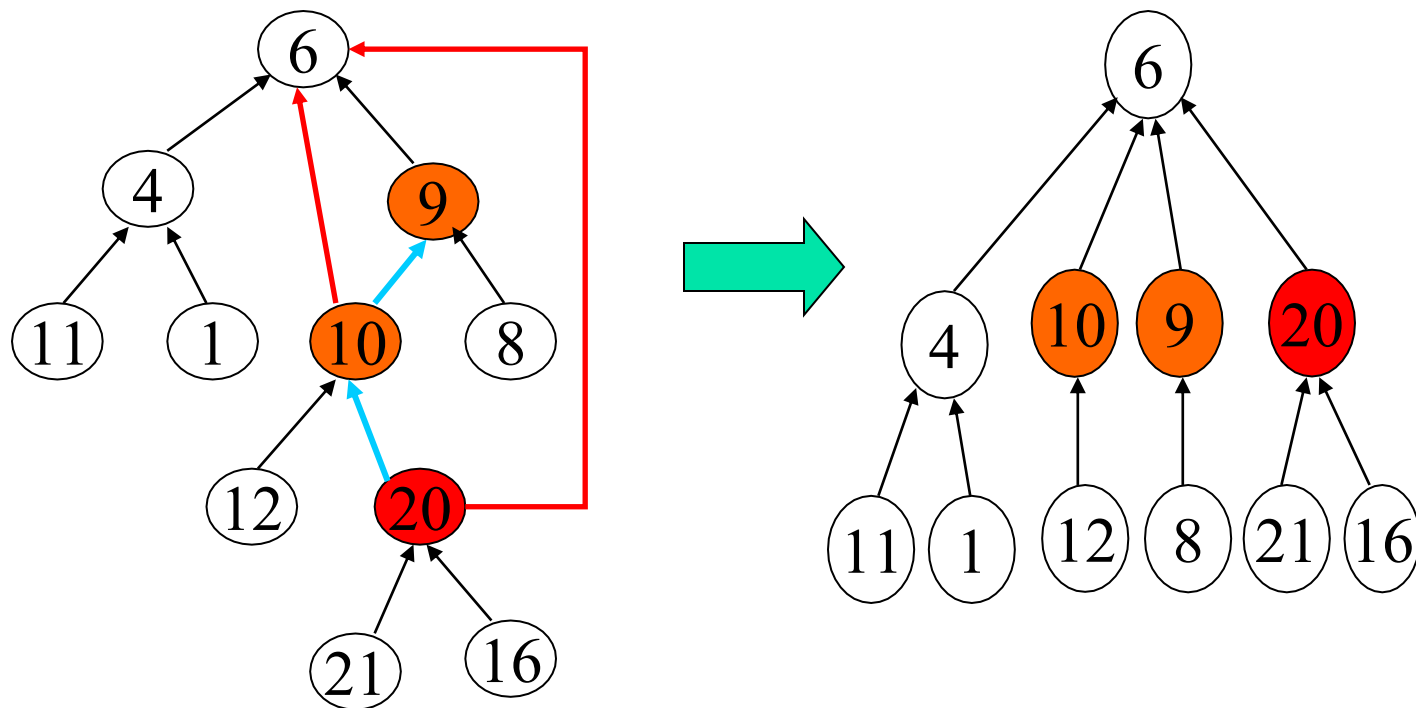
- **思想：** 每次查找的时候，如果路径较长，则修改信息，以便下次查找的时候速度更快
- **步骤：**
 - 第一步，找到根结点
 - 第二步，修改查找路径上的**所有节点**，将它们都指向根结点



带路径压缩的查找算法

```
■ find3(x)
■ {
■     r = x;
■     while (set[r] <> r) //循环结束，则找到根节点
■         r = set[r];
■     i = x;
■     while (i <> r) //本循环修改查找路径中所有节点
■     {
■         j = set[i];
■         set[i] = r;
■         i = j;
■     }
■ }
```

路径压缩示意图





示例一畅通工程(HDOJ-1232)

■ 题目描述：

某省调查城镇交通状况，得到现有城镇道路统计表，表中列出了每条道路直接连通的城镇。省政府“畅通工程”的目标是使全省任何两个城镇间都可以实现交通（但不一定有直接的道路相连，只要互相间接通过道路可达即可）。问最少还需要建设多少条道路？

题目分析

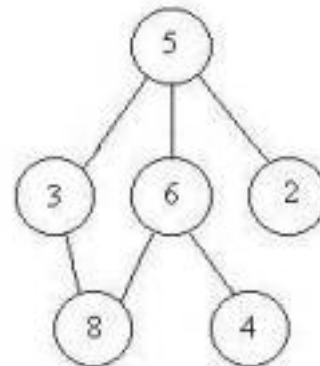
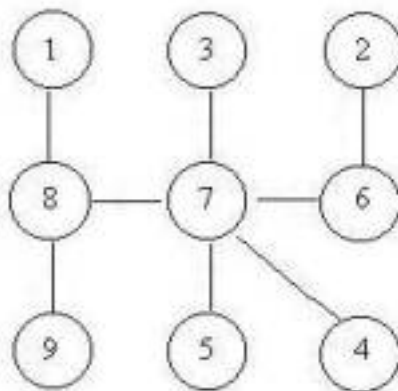
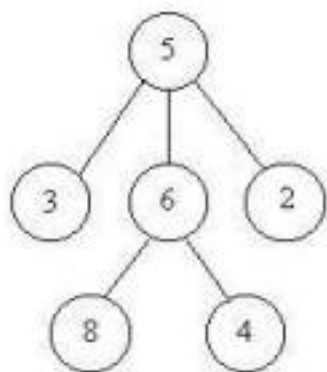
- 最赤裸裸的并查集，无话可说～



示例一小希的迷宫(HDOJ-1272)

■ 题目链接

下面的例子，前两个是符合条件的，但是最后一个却有两种方法从5到达8。



题目分析:

- 该你们来说了~



*Any
question?*





相关练习

- 2008 《ACM Programming》 Exercise(6)_并查集

附加题目:

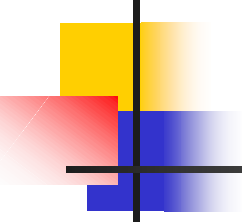
- HDOJ-1558 Segment set
- HDOJ-1811 Rank of Tetris
- HDOJ-1829 A Bug's Life
- HDOJ-1198 Farm Irrigation



附:参考源码(HDOJ-1232)

```
■ #include "stdio.h"
■ int bin[1002];
■ int findx(int x)
■ {
■     int r=x;
■     while(bin[r] !=r)
■         r=bin[r];
■     return r;
■ }
■ void merge(int x,int y)
■ {
■     int fx,fy;
■     fx = findx(x);
■     fy = findx(y);
■     if(fx != fy)
■         bin[fx] = fy;
■ }
```

```
■ int main()
■ {
■     int n,m,i,x,y,count;
■     while(scanf("%d",&n),n)
■     {
■         for(i=1;i<=n;i++)
■             bin[i] = i;
■         for(scanf("%d",&m);m>0;m--)
■         {
■             scanf("%d %d",&x,&y);
■             merge(x,y);
■         }
■         for(count=-1, i=1;i<=n;i++)
■             if(bin[i] == i)
■                 count ++;
■         printf("%d\n",count);
■     }
■ }
```



下一讲:

贪心算法



Welcome to HDOJ

*Thank
You ~*

