

(/apps/redirect?
utm_source=side-
banner-click) x

深夜学算法之Union Find Set：动态连通



kophy (/u/d493d9922395) + 关注

2016.04.30 02:14* 字数 1773 阅读 3549 评论 1 喜欢 6 赞赏 2
(/u/d493d9922395)

1. 前言

并查集（Union Find Set），也称为不相交集数据结构（Disjointed Set Data Structure），两个名字各自概括了这一数据结构的部分特征。简单地讲，并查集维护了一列互不相交的集合S1、S2、S3、...，支持查找（find）与合并（union）两种操作。

- find

找到元素所在的集合，通常返回该集合的代表元（representative）

元素a与元素b是否属于同一个集合，只要判断find（a）与find（b）是否相等

- union

将两个集合合并为一个集合

将元素a与元素b所在的集合合并为一个集合，使用union（a，b）

我的实现在这里 (<https://link.jianshu.com?t=https://github.com/liquidconv/DSAF>)

2. 原理

2.1 基础算法

UnionFindSet用树表示集合，所谓维护一系列互不相交的集合也就是有许多棵树。每棵树的元素属于一个集合，树根的元素就是集合的代表元，所以UnionFindSet实际上维护了一个多棵树构成的森林。

- find(x)就是找x所在的树的树根
- union(x, y)就是合并x和y所在的树

来看UnionFindSet的定义：

```
class UnionFindSet {  
public:  
    UnionFindSet(int n);  
    int Find(int x);  
    void Union(int x, int y);  
private:  
    std::vector<int> parent;  
};
```

与常见的树形数据结构不同，并查集的链接关系不是从parent指向child而是从child指向parent，这个链接信息保存在parent数组里。

1. 若parent[x] == x，则x就是x所在树的树根
2. 若parent[x] != x，则x是parent[x]的子节点

解释完parent的含义，Find的实现方法也就呼之欲出了。



```
int UnionFindSet::Find(int x) {  
    if (parent[x] == x)  
        return x;  
    else  
        return Find(parent[x]);  
}
```

(/apps/redirect?
utm_source=side-
banner-click)

构造函数UnionFindSet(int n)表示最初有n个互不相交的集合，代表元分别为0、1、2、... (n-1)。

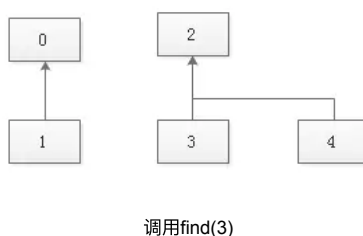
```
UnionFindSet::UnionFindSet(int n) {  
    parent.resize(n);  
    for (int i = 0; i < n; ++i) {  
        parent[i] = i;  
    }  
}
```

要合并两棵树，只要把一棵根节点的parent设为另一棵树的根节点。

```
void UnionFindSet::Union(int x, int y) {  
    int root_x = Find(x);  
    int root_y = Find(y);  
    if (root_x == root_y)  
        return;  
    parent[root_y] = root_x;  
}
```

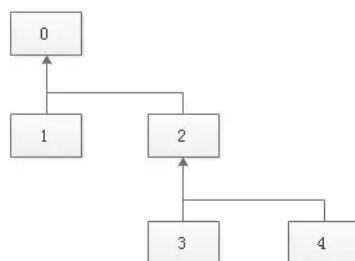
用图形解释find和union中parent的作用，其中child在下层，parent在上层。
3的parent是2，那么调用find(3)时：

- parent[3] = 2，调用find(2)
- parent[2] = 2，返回2



调用union(1, 3)时：

- 调用find(1)，得到root_x = 0
- 调用find(3)，得到root_y = 2
- parent[2] = 0



调用union(1, 3)

2.2 优化合并

并查集就是一组树构成的森林，与通常的搜索树相比只是链接关系从parent->child变成了child->parent，所以也会出现搜索树中的问题。极端情况下n个节点像链表那样构成n层，对于最底层的节点，find复杂度自然是O(n)，而union由于调用find复杂度也会变成O(n)。

既然树的高度影响效率，那么可以设法避免高树出现，也就是本小节的优化合并；也可以设法把高树变矮，也就是下一小节的路径压缩。

优化合并就是优化union操作，简单粗暴地讲，就是合并时永远把矮树作为高树的子树。判断哪棵树比较矮有union-by-size和union-by-height两种做法，union-by-height是在类中添加height数组记录每个节点的高度，union-by-size是在类中添加size数组记录每个节点的子节点数量，两者效果完全相同。我采用union-by-height，感兴趣union-by-size的可以看[这里](https://link.jianshu.com/?t=http://blog.csdn.net/dm_vincent/article/details/7655764) (https://link.jianshu.com/?

t=http://blog.csdn.net/dm_vincent/article/details/7655764)

在类中添加height数组：

```
class UnionFindSet {
public:
    UnionFindSet(int n);
    int Find(int x);
    void Union(int x, int y);
    void Display(void);
private:
    std::vector<int> parent;
    std::vector<int> height;
};
```

修改构造函数UnionFindSet(int n)：

```
UnionFindSet::UnionFindSet(int n) {
    parent.resize(n);
    height.resize(n);

    for (int i = 0; i < n; ++i) {
        parent[i] = i;
        height[i] = 0;
    }
}
```

新的union算法：

(/apps/redirect?
utm_source=side-
banner-click)



```
void UnionFindSet::Union(int x, int y) {
    int root_x = Find(x);
    int root_y = Find(y);

    if (root_x == root_y)
        return;

    if (height[root_x] > height[root_y]) {
        parent[root_y] = root_x;
    } else if (height[root_x] < height[root_y]) {
        parent[root_x] = root_y;
    } else {
        parent[root_y] = root_x;
        ++height[root_x];
    }
}
```

(/apps/redirect?
utm_source=side-
banner-click)

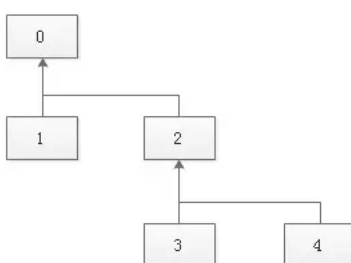
注意只有根节点高度相同时需要更新height，因为两棵树高度不相等时矮的那棵至少比高的矮一层。因此对于有n个节点的并查集，height的最大值为lgn，所以find和union的算法复杂度都至多为 $O(\lg n)$ 。

2.3 路径压缩

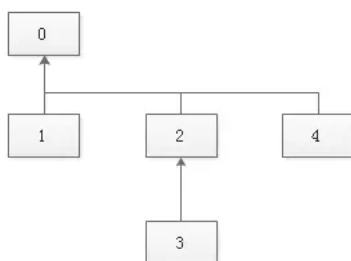
路径压缩就是改造find操作，直接来看代码：

```
int UnionFindSet::Find(int x) {
    if (parent[x] == x)
        return x;
    else {
        int result = Find(parent[x]);
        parent[x] = result;
        return result;
    }
}
```

也就是说在查找时，把查找路径上节点的parent都设为根节点，从而把这棵树「压平」，加速以后的查找。用图形演示find(4)就是：



调用find(4)前



这时find操作也会改变树高，所以height数组里的值就不是准确的树高，而是估计树高（estimated height），或者称为秩（rank），这时做法也成为union-by-rank。

3. 应用

并查集可以用于检测无向图中是否存在环。假设无向图中有v个节点和e条边，用伪代码形式写就是：

```
bool detect_cycle(v, e) {
    u = UnionFindSet(v)
    for (int i = 0; i < e.size; ++i) {

        (p, q) = e(i); // 取得第i条边的两个节点

        if (u.find(p) == u.find(q))
            return true;
        u.union(p, q);
    }
    return false;
}
```

所以可以把并查集用在kruskal最小生成树算法里，实现可以参考这里
(<https://link.jianshu.com?t=http://www.cnblogs.com/venow/archive/2012/08/18/2643984.html>)

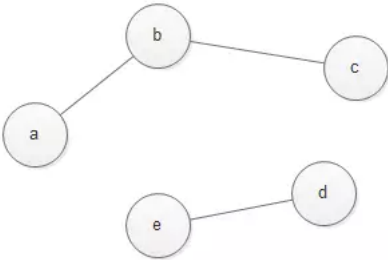
注意与BFS/DFS相比，并查集检测速度快，但只能检测环的存在，不能确定哪些节点构成了环。

4. 参考资料

- Disjoint-set data structure (Wiki) :
https://en.wikipedia.org/wiki/Disjoint-set_data_structure (https://link.jianshu.com?t=https://en.wikipedia.org/wiki/Disjoint-set_data_structure)
- Disjoint-set data structure (Mathblog) :
<http://www.mathblog.dk/disjoint-set-data-structure/> (<https://link.jianshu.com?t=http://www.mathblog.dk/disjoint-set-data-structure/>)

5. 附录：关于代表元

下图是一张典型的无向图，有a、b、c、d、e五个节点：



一张典型的无向图

记** S(x) **表示包含节点x的极大连通子图里节点的集合，显然有：

$S(a) = S(b) = S(c) = \{a, b, c\}$
 $S(d) = S(e) = \{d, e\}$

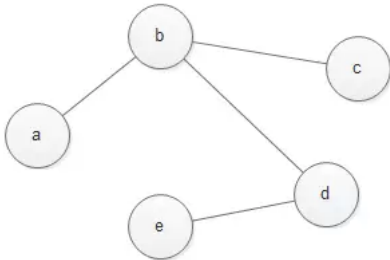
不同的集合有 $\{a, b, c\}$ 和 $\{d, e\}$ 两个，而且它们互不相交。代表元是一个集合里的典型元素，可以从集合里任意选取，比如 $\{a, b, c\}$ 的代表元可以用a，也可以用b或c。由于集合互不相交，所以代表元互不相同，判断「两个元素是否属于同一个集合」等价于判断「两个元素所在集合的代表元是否相等」。

find(x) 就是找到元素x所在集合的代表元。
分别取a、d为两个集合的代表元，那么有：

$find(a) = find(b) = find(c) = a$
 $find(d) = find(e) = d$

b和e不在同一个集合里，相应的有 $find(b) \neq find(e)$ 。

现在我们在原图里添加一条连接b和d的边：



添加边后

发生了什么事情？图上两个互相不连通的部分结合了，这时应该有：

$S(a) = S(b) = S(c) = S(d) = S(e) = \{a, b, c, d, e\}$

union(x, y) 就是将元素x和元素y所在的集合合并成一个集合。
union(b, d)之后，b所在的、以a为代表元的集合，与d所在的、以d为代表元的集合合成了一个大的新集合，将这个新集合代表元取为a，则有：

$find(a) = find(b) = find(c) = find(d) = find(e) = a$

b和e现在在同一个集合里，所以有 $find(b) = find(e)$ 。

小礼物走一走，来简书关注我

赞赏支持



Algorithms (/nb/4241687)

举报文章 © 著作权归作者所有

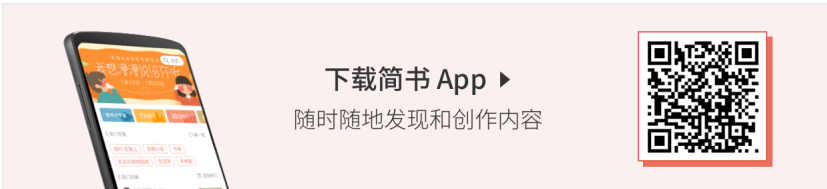


kophy (/u/d493d9922395)

写了 18660 字，被 95 人关注，获得了 137 个喜欢 (/u/d493d9922395)

+ 关注

今でもあなたはわたしの光 <https://github.com/kophy>



(/apps/redirect?utm_source=note-bottom-click)



登录 (/sign-in?utm_source=desktop&utm_medium=not-signed-in-comment-form)

1条评论 只看作者 按时间倒序 按时间正序

gold_paladin (/u/54eaacbc0f8f)
2楼 · 2017.12.29 05:22
(/u/54eaacbc0f8f)
谢谢解答

赞 回复

被以下专题收入，发现更多相似内容

- 首页投稿（暂停... (/c/bDHhpK?utm_source=desktop&utm_medium=notes-included-collection)
- 程序员 (/c/NEt52a?utm_source=desktop&utm_medium=notes-included-collection)
- IT共论 (/c/1113b792c0dc?utm_source=desktop&utm_medium=notes-included-collection)
- IT相关 (/c/94294909cd08?utm_source=desktop&utm_medium=notes-included-collection)
- 编程学习 (/c/d6f06c8a38d0?utm_source=desktop&utm_medium=notes-included-collection)
- 算法 (/c/8925e1fdeb27?utm_source=desktop&utm_medium=notes-included-collection)

推荐阅读 更多精彩内容 > (/)

秋日行 (/p/b3ea1a791c6e?utm_campaign=maleskine&utm_content=no...

人生百岁一抹云，天地万物皆微尘。八风不动通三界，四维洞察释六轮。山中浮彩过风减，岭上飞花经霜贫。黄花盈袖幽曲径，红叶举头照行人。白露寒霜金风壮，秋雨缄泪相思纯。蝉歌蛩...

慕愚堂 (/u/0fb7967afeac?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

《三千繁花只为你》第五十二章 大战（下） (/p/21199b8ccde5?utm_campaign=maleskine&utm_content=note&utm_source=side-banner-click) (/p/21199b8ccde5?utm_campaign=maleskine&utm_content=note&utm_source=side-banner-click)

东华仙遁到魔族时，双方还在激战，炙焱结了一道屏障，躲在里面观战，却不出手。白家的众位上神和夜华折颜，此时也立在一旁，和炙焱对峙...

小小的琳琅 (/u/da0644031297?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation) (/apps/redirect?utm_source=side-banner-click)

如果性侵门主角是奶茶..... (/p/e324900371b9?utm_campaign=maleskine&utm_content=note&utm_source=recommendation) (/p/e324900371b9?utm_campaign=maleskine&utm_content=note&utm_source=recommendation)

文 | 蔡垒磊 大佬也有栽的时候。昨天看到刘强东性侵的消息时，我一点都不奇怪，不说大佬们被塞外围的可能性比普通人高很多，就单说被人认...

蔡垒磊 (/u/4aa0ab9ef3fb?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

程序员生存图鉴：接私活、创业、自由职业，哪种才是... (/p/b3f3e520af75?utm_campaign=maleskine&utm_content=note&utm_source=recommendation) (/p/b3f3e520af75?utm_campaign=maleskine&utm_content=note&utm_source=recommendation)

前阵子有个叫「隐形贫困人口」的词很火，指的是那些看上去生活滋润，实际上并不富裕的人。而与之相反的恐怕就是「隐形富裕人口」了。在...

陈序元 (/u/aa054c153baf?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

500万网友支持张雨绮：我找男人是不行，但老娘对爱... (/p/512c3b21ec61?utm_campaign=maleskine&utm_content=note&utm_source=recommendation) (/p/512c3b21ec61?utm_campaign=maleskine&utm_content=note&utm_source=recommendation)

文/麦大人 江湖我张姐，人狠话不多；怒扇汪小菲，脚踹王全安。昨日砍巴元，今天就离婚；一刀断过往，一刀敬明天。昔恋她颜值，今迷她果...

麦大人 (/u/2b3ad4f2a058?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

(/p/9bc86a6bbbbc?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation) (/p/9bc86a6bbbbc?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)




LeetCode/LintCode ReviewPage 题解-总结 (/p/9bc86a6bbbbc?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation) (/p/9bc86a6bbbbc?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

背景 一年多以前我在知乎上答了有关LeetCode的问题，分享了一些自己做题目的经验。张土汪：刷leetcode是什么样的体验？慢慢有一些赞和感谢，备受鼓舞，于是我把所做过的题目用一个script跑...

 张土汪 (/u/0e6b059ad57e?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)


百战程序员V1.2——尚学堂旗下高端培训_ Java1573题 (/p/49ad52bd5405.../p/49ad52bd5405...)

百战程序员_ Java1573题 QQ群：561832648489034603 掌握80%年薪20万掌握50%年薪10万 全程项目穿插，从易到难，含17个项目视频和资料持续更新，请关注www.itbaizhan.com 国内最牛七星...

 Albert陈凯 (/u/185a3c553fc6?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)


基础知识 (/p/be06441213cc?utm_campaign=maleskine&utm_content=.../p/be06441213cc?utm_campaign=maleskine&utm_content=...) (/p/be06441213cc?utm_campaign=maleskine&utm_content=.../p/be06441213cc?utm_campaign=maleskine&utm_content=...)

1. 关于诊断X线机准直器的作用，错误的是（ ）。(6.0 分) A. 显示照射野 B. 显示中心线 C. 屏蔽多余原射线 D. 吸收有用射束产生的二次射线 E. 把病人接受剂量限制到最低 参考答案: D 本题解释 2# 2...

 我们村我最帅 (/u/19ebb9b604a5?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

偶成 (/p/0ca9ae7aa5b9?utm_campaign=maleskine&utm_content=note...

珠帘卷别意， 琴筝奏悲乐。晚风又吹面， 乡情把念牵。

 苍穹橙子 (/u/035124f59067?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)


(/p/12b2501bcaf3?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

遊藝- 旅遊的藝術在於發現生活在別處的味道 (/p/12b2501bcaf3?utm_cam...

「游憶- Recall Dubai International Airport」 迪拜机场最讓我驚喜是當我手機在公共區域充電而我本人已困得睡着后的一個小時醒來手機依然在原位置充電，感嘆這裡遊客的素養，至少我沒遇到小偷。...

 CulturalAngel (/u/6464a9cec1f0?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)


(/p/81cbbba5f505?



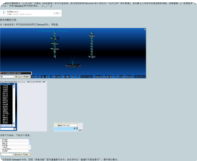
utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

关于孩子便便的秘密 你知道几个 (/p/81cbbba5f505?utm_campaign=male...

大家好，我是大便君，先别露出嫌弃的脸，毕竟我也曾经是美味的食物！而且在你肚子里溜达了一圈，我发现了你的很多秘密！所以请你冲走我之前请看我一眼，我会告诉你很多健康问题... 1、目...

 育儿趣学园 (/u/935ea26289e1?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)


(/p/b89e67f7377a?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

T20_38 解压BrainZip File文件、Thebrain间复制粘贴想法或合并Theb... (/...

(2016-12-05-周一 20:51:59)

 菜五 (/u/0a23b54b7b4d?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

