

Union-Find Structure

yanQval

IIIS, Tsinghua

2019 年 7 月 18 日

Union-Find Problem

Union-Find Problem

Definition

给定一个包含 n 个元素的集合 $\{1, 2, \dots, n\}$

初始时每个元素单独组成一个集合, $\{1\}, \{2\}, \dots, \{n\}$

操作序列由合并操作和查询操作混合而成

合并操作将两个集合合并, 任意元素任意时刻只存在于一个集合内

查询操作查询一个元素当前位于哪个集合内

Disjoint sets

Disjoint sets

维护若干不交的集合

Disjoint sets

维护若干不交的集合
每个集合拥有一个代表元素

Disjoint sets

维护若干不交的集合

每个集合拥有一个代表元素

支持三种操作：

- 新建集合， $\text{MakeSet}(x)$
- 合并集合， $\text{Union}(x,y)$
- 查询， $\text{Find}(x)$

Disjoint sets

维护若干不交的集合

每个集合拥有一个代表元素

支持三种操作：

- 新建集合， $\text{MakeSet}(x)$
- 合并集合， $\text{Union}(x,y)$
- 查询， $\text{Find}(x)$

通常查询操作并不要求对特定的集合返回一个特定的值，只要求 $\text{find}(i)$ 与 $\text{find}(j)$ 返回相同的值当且仅当 i 和 j 同处于一个集合内。

Trivial implementation

Trivial implementation

考虑任意时刻对于每个元素直接维护它所在集合的代表元。
集合建立的时候代表元为自身。每当执行合并 x 和 y 的时候，暴力修改其中一个集合的所有元素的代表元为另外一个集合的代表元。

Trivial implementation

考虑任意时刻对于每个元素直接维护它所在集合的代表元。
集合建立的时候代表元为自身。每当执行合并 x 和 y 的时候，暴力修改其中一个集合的所有元素的代表元为另外一个集合的代表元。

假设操作共有 m 次。
查询操作单次复杂度 $O(1)$ ，合并操作单次复杂度为 $O(n)$ ，考虑到有效的合并操作只有 $n - 1$ 次，该实现方法总复杂度 $O(n^2 + m)$ 。

A simple improvement

A simple improvement

考虑一个简单的优化，维护每个集合的大小，每次合并的时候，暴力修改较小的集合内所有元素所记录的代表元。

A simple improvement

考虑一个简单的优化，维护每个集合的大小，每次合并的时候，暴力修改较小的集合内所有元素所记录的代表元。

查询操作没有发生变化。

考虑当一个元素的代表元被修改的时候，若之前所在的集合大小为 x ，则合并后的集合大小至少为 $2x$ ，i.e., 经过至多 $\log_2 n$ 次修改代表元操作之后，该元素就被合并到了最大的集合中。该实现方法总复杂度 $O(n \log n + m)$ 。

Up-Trees

Up-Trees

使用森林结构同样可以表示一个集合。

Up-Trees

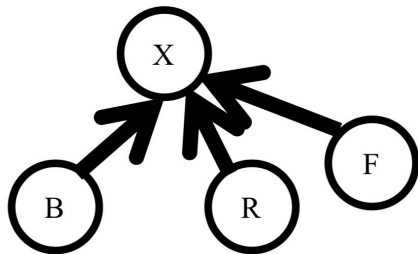
使用森林结构同样可以表示一个集合。

一棵树表示一个集合，自然地使用根节点作为表示元。

Up-Trees

使用森林结构同样可以表示一个集合。

一棵树表示一个集合，自然地使用根节点作为表示元。



集合 $\{B, F, R, X\}$ ，其中代表元为 X 。

Operations

Operations

查询操作:

查询一个节点所在树的根 \Rightarrow 维护每个节点的父亲

合并操作:

将两棵树合并为一棵树

直接将其中一棵树的根的父亲设置为另外一棵树的根

Code

```
for(int i=1;i<=n;i++)fa[i]=i;
```

```
int find(int x){  
    while(fa[x]!=x)x=fa[x];  
    return fa[x];  
}
```

```
int union(int x,int y){  
    x=find(x);y=find(y);  
    if(x!=y)fa[x]=y;  
}
```

Complexity

Complexity

显然复杂度集中在查询操作上，查询操作的复杂度直接与深度相关。

Complexity

显然复杂度集中在查询操作上，查询操作的复杂度直接与深度相关。

考虑最坏情况：



依次执行：UNION(D,E), UNION(C,D), UNION(B,C), UNION(A,B)。

Two Heuristics

Two Heuristics

- 按权合并或按秩合并

Two Heuristics

- 按权合并或按秩合并
- 路径压缩

Two Rules

Two Rules

按秩合并：

在合并两棵树时，令高度较低的一棵树成为另外一棵树的子树。

Two Rules

按秩合并:

在合并两棵树时, 令高度较低的一棵树成为另外一棵树的子树。

按权合并:

在合并两棵树时, 令节点数较少的一棵树成为另外一棵树的子树。

Complexity

Complexity

按照两种方法实现的并查集均只有 $O(\log_2 n)$ 的深度。
关于按权合并的证明类似前文。

Complexity

按照两种方法实现的并查集均只有 $O(\log_2 n)$ 的深度。

关于按权合并的证明类似前文。

考虑只有两颗深度相同的树合并时才会增加深度，归纳证明深度为 i 的树大小至少有 $2^i - 1$ 。

Path Compression

Path Compression

每当我们执行查询操作的时候，我们将沿途路过的所有点的父亲全部直接置为根。

```
int find(int x){  
    if(fa[x]==x)return x;  
    return fa[x]=find(fa[x]);  
}
```

路径压缩的并查集的复杂度也为 $O(\log_2 n)$ ，证明超出提高组知识。

Combine Height and Path Compression

Combine Height and Path Compression

如果我们同时使用按秩合并和路径压缩会怎么样？

Combine Height and Path Compression

如果我们同时使用按秩合并和路径压缩会怎么样?

Definition for Ackermann function

$$A(m, n) = \begin{cases} n + 1 & m = 0 \\ A(m - 1, 1) & m > 0, n = 0 \\ A(m - 1, A(m, n - 1)) & m > 0, n > 0 \end{cases}$$

反阿克曼函数 $\alpha(n) = \min\{k : A(k, 1) \geq n\}$.

同时使用两者的并查集，对大小为 n 的集合执行 m 次操作的复杂度为 $O(m\alpha(n))$.

Is it optimal?

Is it optimal?

Tarjan 于 1975 年证明了该算法 $O(m\alpha(n))$ 的复杂度，同时证明这也是该算法的复杂度下界。

Fredman 和 Saks 于 1989 年证明了 UNION-FIND 问题的复杂度下界就是 $\Omega(m\alpha(n))$ 。

[NOI2015] 程序自动分析

给定 n 个等式或不等式，如 $x_i = x_j$ 或 $x_i \neq x_j$ ，求是否存在合法的赋值方案满足所有的式子。

$n \leq 100000$

关押罪犯

有 n 名罪犯，有 m 对关系 (a, b, c) 表示罪犯 a 和 b 之间有矛盾值 c 。现在有两间监狱要关押所有犯人，如果两个犯人同处一个监狱，他们就会爆发矛盾。最小化爆发的最大的矛盾。

$$n, m \leq 10^5$$

[JSOI2008] 星球大战

给定一张 n 个点， m 条边的无向图，接下来按某种顺序删掉所有边，询问每次删掉边之后的联通块的数量。允许离线。

$m \leq 200000, n \leq 2m$

动态图

给定一张 n 个点的空图，共有 m 次操作，每次操作加一条边或删除一条已经存在的边，每次操作后输出当前图的联通块的数量。允许离线
 $n, m \leq 10^5$

[NOI2001] 食物链

有三种动物 ABC ，满足 $A \rightarrow B \rightarrow C \rightarrow A$ ，其中箭头表示前者吃后者。现有 n 个动物，但并不知道他们的种类，现在有依次的 K 句话，每句话是如下的两个形式之一

- x 和 y 是同类
- x 吃 y

如果一句话和前面的所有话没有冲突，那么就是真话，否则是假话，我们之后也不考虑它了，求有多少假话。

[HNOI2005] 狡猾的商人

账本上记录了 n 个月的盈利情况，盈利有正有负。你偷看了 m 次，每次偷看的时候你看到了一个区间内的总盈利。判断账本是否是假的。

$n, m \leq 100000$

疯狂的馒头

有一个长度为 N 的数列，初始全部为 1，有 M 次操作，每次操作把 $[l_i, r_i]$ 的数全部变成 k_i ，求最终的数列。

$$n, m \leq 10^6$$

遥远的星系

给定一个 n 个点的图，图中存在 m 条边 (u, v, w) ，表示如果从 u 走到 v 要花费恰好 w ，反之从 v 到 u 花费 $-w$ 。有 q 次询问，每次询问是否能从 u 走到 v 花费恰好 w 。每条边允许经过很多次。

$$n, m, q \leq 10^6$$

[SCOI[2016] 萌萌哒

一个长度为 N 的十进制整数 $S_1 S_2 \dots S_N$ ，有 M 条限制，每条限制要求两个区间完全相同。求符合条件的数的个数。
 $N, M \leq 10^5$ 。