

电子科技大学

## 基于不相交集的问题求解报告

课程：《ACM-ICPC 算法与程序设计》

学院：微电子与固体电子学院

学号：2016030102010

选课号：

教师：杨鹏

姓名：傅宣登

论文成绩：

教师签名：

## 摘要

不相交集是一种常用的数据结构。这种数据结构实现简单，操作迅速。

简单地说，不相交集可用于快速查询一堆元素是否属于同一集合。鉴于在有些问题中，这种关系相当隐秘，本文采用另一种更理论的方式来描述这种关系。

本文讨论了实现不相交集的基础数据结构：数组、红黑树、树。本文采用树来实现不相交集。接着给出了朴素的求并、查询算法，然后用所谓的路径压缩优化了查询算法。

最后给出了一道例题，并写出了完整的求解报告。

**关键字：** 不相交集 算法 数据结构 程序设计

## 正文

### 一、不相交集 ADT

不相交集（并查集）是描述解决等价问题的一种有效数据结构。这种数据结构实现起来非常简单，而且每种操作只需要常数平均时间。许多算法中都用到了不相交集，例如 Kruskal 算法和 Prim 算法等。

#### 1.1 等价关系

若对每一对元素  $(a, b)$ ,  $a, b \in S$ ,  $aRb$  要么为 true 要么为 false, 则称在集合  $S$  上定义关系  $R$ 。如果  $aRb$  是 true, 那么我们说  $a$  与  $b$  有关系。

**定义 1 (等价关系)** 等价关系是满足下列三个性质的关系  $R$ :

- (自反性) 对于所有的  $a \in S$ ,  $aRa$ ;
- (对称性)  $aRb$  当且仅当  $bRa$ ;
- (传递性) 若  $aRb$  且  $bRc$ , 则  $aRc$ 。

这里有几个例子。

如果两个城市位于同一个国家, 那么定义他们是有关系的。容易验证这是一个等价关系。如果能够通过公路从城镇  $a$  旅行到  $b$ , 则设  $a$  与  $b$  有关系。如果所有的道路都是双向行驶的, 那么这种关系也是一个等价关系。

#### 1.2 动态等价性问题

给定一个等价关系 “ $\sim$ ”, 一个自然的问题是对任意的  $a$  和  $b$ , 确定是否  $a \sim b$ 。如果将等价关系存储为一个二维布尔数组, 那么当然这个工作可以以常数时间完成。问题在于这种关系的定义通常不明显甚至相当隐秘。

例如, 设在 5 个元素的集合  $\{a_1, a_2, a_3, a_4, a_5\}$  上定义一个等价关系。此时存在 25 对元素, 他们的每一对要么有关系要么没有关系。然而, 信息  $a_1 \sim a_2, a_3 \sim a_4, a_4 \sim a_2, a_1 \sim a_5$  意味着每一对元素都是有关系的。我们需要一个能快速判断出这些关系的数据结构。

**定义 2 (等价类)** 一个元素  $a \in S$  的等价类是  $S$  的一个子集, 它包含所有与  $a$  有关系的元素。

显然, 等价类形成对  $S$  的一个划分:  $S$  的每一个元素恰好出现在一个等价类中。这样, 为确认是否  $a \sim b$ , 我们只需验证  $a$  和  $b$  是否都在同一个等价类中。

输入数据最初是由  $N$  个集合的类，每个集合含有一个元素。初始的描述是所有的关系均为假（自反的关系除外）。每个集合都有一个不同的元素，从而  $S_i \cap S_j = \emptyset$ ；即所有集合不相交。

此时，我们定义两种运算。第一种是 Find，它返回包含给定元素的集合（即等价类）的名字。第二种是添加关系。如果我们想要添加关系  $a \sim b$ ，那么我们首先要看是否  $a$  和  $b$  已经有关系。这可以通过对  $a$  和  $b$  执行 Find 并检查它们是否在同一个等价类中来完成。如果它们不在同一个等价类中，那么我们使用求并运算 Union，这种运算把含有  $a$  和  $b$  的两个等价类合并成一个新的等价类。从集合的观点看， $\cup$  的结果是建立一个新集合  $S_k = S_i \cup S_j$ ，并去掉原来两个集合而保持所有的集合的不相交性。由于这个原因，常常把这项工作的算法叫做不相交集的 Union/Find 算法，不相交集也叫并查集。

## 二、不相交集的实现

### 2.1 基本数据结构

### 2.2 并查算法

## 三、例题：TROY Query

这是 Codeforces Hello 2015 (Div. 2) 的 F 题<sup>1</sup>。

## 四、符号说明

符号	意义
D	木条宽度 (cm)
L	木板长度 (cm)
W	木板宽度 (cm)
N	第 $n$ 根木条
T	木条根数
H	桌子高度 (cm)
R	桌子半径 (cm)
R	桌子直径 (cm)

<sup>1</sup><http://codeforces.com/gym/100571/problem/F>

## 五、问题分析

### 5.1 问题三分析

题目要求制作软件的意思就是客户给定折叠桌高度、桌面边缘线的形状大小和桌脚边缘线的大致形状,将这些信息输入程序就得到客户想要的桌子。我们在求解最优设计加工参数时,自行给定桌面边缘线形状(椭圆、相交圆等),桌脚边缘线形状,折叠桌高度,应用第二问的非线性规划模型,用 MATLAB 软件绘制折叠桌截面图,得到自己设计的创意平板折叠桌。

问题三流程图:

## 六、参考文献

- [1] M. A. Weiss. 数据结构与算法分析:C 语言描述 (原书第 2 版)[M]. 冯舜玺译. 北京: 机械工业出版社, 2004:199-207.
- [2] 胡伟.  $\text{\LaTeX}$  完全学习手册. 清华大学出版社, 北京, 2011.
- [3] L. Lamport.  $\text{\LaTeX}$ : *A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, 2nd edition, 1994.
- [4] F. Mittelbach, M. Goossens, J. Braams, D. Carlisle, and C. Rowley. *The \text{\LaTeX} Companion*. Addison-Wesley, Reading, Massachusetts, 2nd edition, 2004.

## 附录 A Hello 源代码

```
1  /* Program Hello World. */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      return !printf("Hello World\n");
7  }
```