《算法竞赛入门到进阶》: 勘误和改进

出版社:清华大学出版社作者:罗勇军郭卫斌

最近更新: 2019.8.28, 第3次印刷

本书在多次重印过程中,进行了持续改进。本文记录了这些改进的细节,包括两方面的内容:

- (1) 勘误。本书的印刷或内容错误,每次新印刷时,会修改新发现的问题。
- (2)新内容。增加的新内容。

本文下载地址:

- (1) https://github.com/luoyongjun999/code
- (2) QQ 群: 567554289

请读者多提意见,非常感谢! 联系方式: QQ 15512356,邮箱 15512356@qq.com

一、改进历史

时间	页码	改正内容
勘误: 2019.7	55 页	把(c, r)改成(r, c)
第2次印刷已改	294 页	把k改成n
勘误: 2019.8	23 页	"在排列问题中,如果要求输出所有的全排列"
第3印已改	109 页	图 6.7(b), 3 改为 13
新内容:第3印		增加 31 个新视频
	210 页	" 总复杂度是 O(n(log ₂ n) ²) "
勘误: 待改正	254 页	把"直到所有边都在T中"改为"直到所有点都在T
		中"
	24 页	把"map:一对多映射"改为"map:一对一映射"
新内容: 待加入	316页	第 12 章-表 12.1.mp4

二、勘误细节

- 1、第2次印刷已改正部分
- (1) 55页, 把(c,r)改成(r,c)

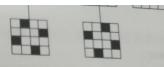


图 4.8 四皇后问题的搜索树

该图用 BFS 和 DFS 都能实现。前文说过, DFS 的代码比 B 来解决。

关键问题:在扩展结点时如何去掉不符合条件的子结点? 设左上角是原点(0,0),已经放好的皇后的坐标是(i,j),不 皇后的坐标是(c,r),它们的关系如下:

- (1) 横向,不同行: $i\neq r$ 。
 - (2) 纵向,不同列: $j\neq c$ 。
- (3) 斜对角: 从(i,j)向斜对角走 a 步,那么新坐标(r,c) 有 4 右上(i+a,j-a)、左下(i-a,j+a)、右下(i+a,j+a),综合起皇后的位置不能放在斜线上,需满足 $|i-r|\neq |j-c|$ 。
- (2) 294页, 把k改成n。

```
//0: 线段在圆内
      if(sgn(dst - C.r) < 0) return 0;
                                         //1: 线段和圆相切
      if(sgn(dst - C.r) == 0) return 1;
                                         //2: 线段在圆外
      return 2;
   5. 直线和圆的交点
   求直线和圆的交点可以按图 11.20 所示,先求圆心 c 在直线
上的投影q,再求距离d,然后根据r和d求出长度k,最后求出
两个交点 p_a = q + n * k、p_b = q - n * k,其中义是直线的单位
向量。
    //pa、pb是交点.返回值是交点的个数
                                                   图 11.20
    int Line_cross_circle(Line v, Circle C, Point &pa, Point &pb) {
       if(Line_circle_relation(v, C) == 2) return 0; //无交点
                                          //圆心在直线上的投影占
```

- 2、第3印已改正部分
- (1) 23页,修改。

每个数操作一次,所以总复杂度是 $O(n\log_2 n)$ 。用分治法思想实现的快速排序算法排序算法的复杂度就是 $O(n\log_2 n)$ 。

5. $O(n^2)$

一个两重循环的算法,复杂度是 $O(n^2)$ 。例如冒泡排序是典型的两重循环。类杂度有 $O(n^3)$ 、 $O(n^4)$ 等。

6. $O(2^n)$

一般对应集合问题,例如一个集合中有 n 个数,要求输出它的所有子集,子集有

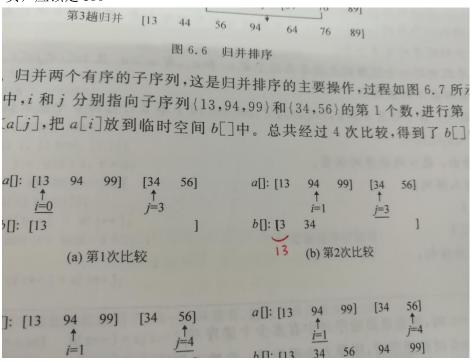
在集合问题中,如果要求按顺序输出所有的承集,那么复杂度就是 O(n!)。

把上面的复杂度分成两类: ①多项式复杂度,包括 O(1)、O(n)、 $O(n\log_2 n)$ 、 $O(n\log_2 n)$ 、 $O(n\log$

如果一个算法是多项式复杂度,称它为"高效"算法;如果一个算法是指数复杂 它为"低效"算法。可以这样通俗地解释"高效"和"低效"算法的区别:多项式复杂度 随着规模 n 的增加可以通过堆叠硬件来实现,"砸钱"是行得通的;而指数复杂度的 加硬件也无济于事,其增长的速度超出了人们的想象力。

竞赛题目一般的限制时间是 1s,对应普通计算机的计算速度是每秒千万次级述的时间复杂度可以换算出能解决问题的数据规模。例如,如果一个算法的复

(3) 109页,应该是13。



3、待改正部分

(1) 210页: 待改正。

上面的程序用到的 sort()实际是快速排序,每一步排序的复 log_2n 个步骤,总复杂度是 $O(nlog_2n)$ 。虽然已经很好了,不过 法——基数排序,总复杂度只有 $O(nlog_2n)$ 。在下一节的问题 hdu 和基数排序两种方案的倍增法程序,执行时间分别是 1000ms 和 log_2n

3. 基数排序

、一日出比较高

(2) 254 页: "所有边"应该是"所有点"。待改正。

庄都连通(但不一定有直接的公路相连,只安肥问妆~~~ 度最小?请计算最小的公路总长度。

图的两个基本元素是点和边,与此对应,有两种方法可以构造最小生成构法都基于贪心法,因为 MST 问题满足贪心法的"最优性原理",即全局最优定prim 算法的原理是"最近的邻居一定在 MST 上",kruskal 算法的原理是"最MST 上"。

- (1) prim 算法:对点进行贪心操作。从任意一个点 u 开始,把距离它最到 T 中;下一步,把距离 $\{u,v\}$ 最近的点 w 加入到 T 中;继续这个过程,直
- (2) kruskal 算法:对边进行贪心操作。从最短的边开始,把它加入到了边中找最短的边,加入到 T中;继续这个过程,直到所有边都在 T中。

在这两个算法中,重要的问题是判断圈。最小生成树显然不应该有圈,小"了。所以,在新加入一个点或者边的时候要同时判断是否形成了圈。

10.10.1 prim 算法

图 10.25 说明了 prim 算法的步骤。设最小生成树中的 与的 是 的 是 A B

(3) 24 页: "map: 一对多映射"应该是"map: 一对一映射"。待改正。

2. 关联式容器

关联式容器包括 set、multiset、map、multimap 等。

- set: 集合,快速查找,不允许重复值。
- · multiset: 快速查找,允许重复值。
- · map: 一对多映射,基于关键字快速查找,不允许重复
- · multimap: 一对多映射,基于关键字快速查找,允许重
- (4) 32 页, hdu 2094 题,应该要求测试数据是连通图,这样更严谨一些。

例如:

4

a b

b c

са

d e

这样的数据,图不是连通的。判 No 更符合逻辑。