

《算法竞赛入门到进阶》：勘误与改进

出版社：清华大学出版社

作者：罗勇军 郭卫斌

联系方式：QQ 15512356，QQ 群：567554289

最近更新：2019.11.5，第 4 次印刷。

在扉页中可以查到是第几次印刷。

本书在多次重印过程中，进行了持续勘误和改进。本文记录了这些细节，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：

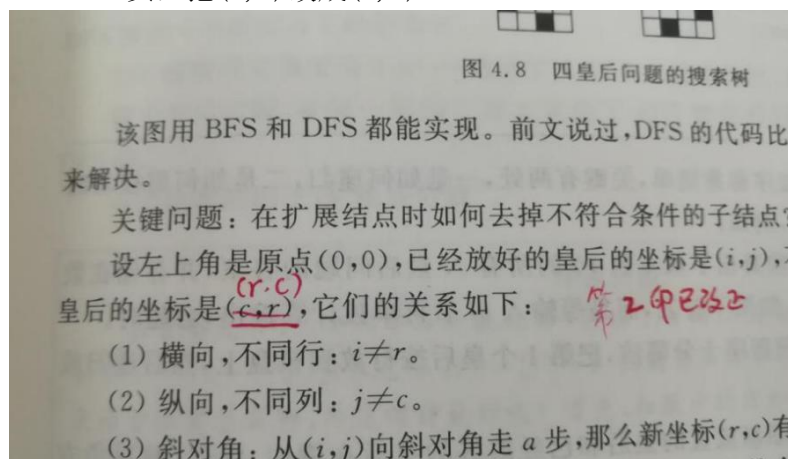
(1) <https://github.com/luoyongjun999/code>

(2) QQ 群：567554289 群文件

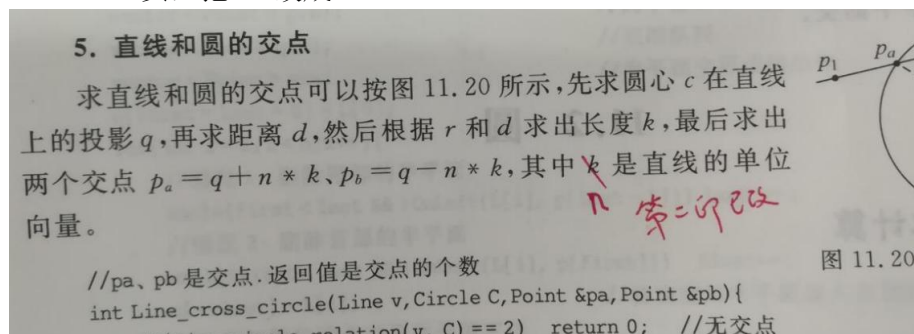
1、第 2 次印刷（2019.8）

以下已经改正：

(1) 55 页，把 (c, r) 改成 (r, c)



(2) 294 页，把 k 改成 n 。



2、第 3 次印刷（2019.9）

改进：前面 2 印是 120 分钟视频；从第 3 印开始，增加到 350 分钟视频。

下面是修改：

(1) 23 页，修改一句。

6. $O(2^n)$

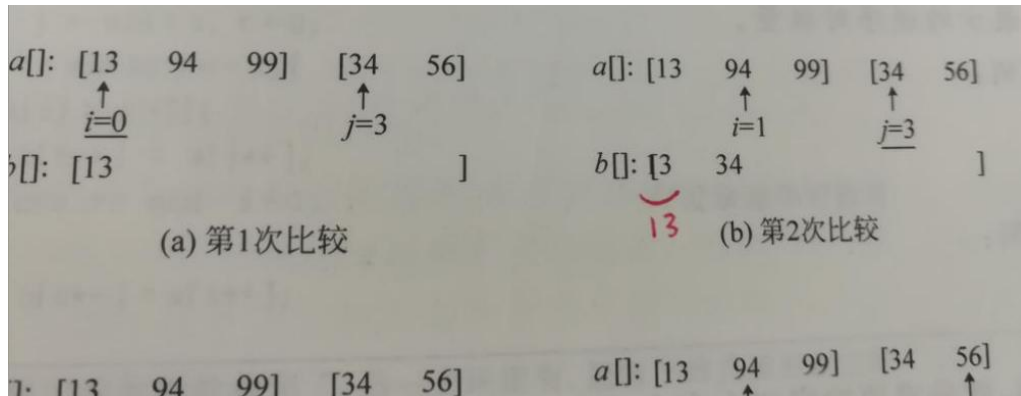
一般对应集合问题,例如一个集合中有 n 个数,要求输出它的所有子集,子集有

7. $O(n!)$

在集合问题中,如果要求按顺序输出所有的子集,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 $O(1)$ 、 $O(n)$ 、 $O(n\log_2 n)$ 、 $O(n^k)$ 中 k 是一个常数;②指数复杂度,包括 $O(2^n)$ 、 $O(n!)$ 等。

(3) 109 页,应该是 13。



3. 第4次印刷 (2019.11)

以下内容已经修改:

(1) 21 页。把 $i=\text{MAXN}$ 改为 $i=\text{MAXN}-1$

第2章 算法复杂度

```

int t;
scanf("%d", &t);           //此题数据多,如果
a[500000 + t] = 1;         //数字 t, 登记在 5
}
for(int i = MAXN; m > 0; i--)
    if(a[i]){
        if(m > 1) printf("%d ", i - 500000);
        else      printf("%d\n", i - 500000);
        m--;
    }
return 0;
    
```

(2) 24 页。把“map: 一对多映射”改为“map: 一对一映射”。

2. 关联式容器

关联式容器包括 set、multiset、map、multimap。

- set: 集合, 快速查找, 不允许重复。
- 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。
- multimap: 一对多映射, 基于关键字快速查找。

(3) 27 页。j++ 没对齐, 往后退几格对齐。

```
int j = 0;
for(int i = 0; i < 2 * n; i++){
    if(!(i % 50) && i) cout << endl;
    if(j < table.size() && i == table[j]){
        j++;
        cout << "G";
    }
    else
        cout << "B";
}
```

(4) 32 页。第 6 行括号里面少一个 k

算法竞赛入门到

```
A.clear(); //清空
A.empty(); //判断
A.size(); //返回
A.find(k); //返回
A.lower_bound(k); //返回
A.upper_bound(k); //返回
```

下面用一个例子来说明 set 的应用。

hdu 2094 “产生”

有一群人打乒乓球比赛, 两两配对比赛。

(5) 48 页。删除公式末尾的 [1]

产生的全排列按字典序排序,第 X 个排列的计算公式如下:

$$X = a[n] \times (n-1)! + a[n-1] \times (n-2)! + \dots + a[i] \times (i-1)! + \dots + a[2] \times 1! + a[1] \times 0!$$

未出现的元素排在第几个(从 0 开始),并且有 $0 \leq a[i] < i$ ($1 \leq i \leq n$)。反过程是康托逆展开: 某个集合的全排列,输入一个数字 k ,返回第 k

(6) 48 页。把倒数第 2 行的“ $a[i]$ 为当前未出现的元素排在第几个”改为“ $a[i]$ 表示原数的第 i 位在当前未出现的元素中排在第几个”

把一个集合产生的全排列按字典序排序,第 X 个排列的计算

$$X = a[n] \times (n-1)! + a[n-1] \times (n-2)! + \dots + (i-1)! + \dots + a[2] \times 1! + a[1] \times 0!$$

其中, $a[i]$ 为当前未出现的元素排在第几个(从 0 开始),并且有 $0 \leq a[i] < i$ ($1 \leq i \leq n$)。上述过程的反过程是康托逆展开: 某个集合的全排列,输入一个数字 k ,返回第 k 个排列。

• 48 • 表示原数的第 i 位在

(7) 59 页。删除 `int t;`

```
return false;
}
int main(){
int t;
while(cin >> n && n){
    int depth;
    for(depth = 0 ; ; depth++){
        val[pos = 0] = 1;
        if(ida(0, depth)) break;
    }
}
```

(8) 99 页。

最下面的脚注,原来的说法是错误的。现在改为:

一个简单的判断标准是,面值是整数 c 的幂, c^0, c^1, \dots, c^k , 其中 $c > 1, k \geq 1$, 可以用贪心法。例如以 2 的倍数递增的 1、2、4、8 等,这样的面值就符合条件。

那么,如何判断一个题目能用贪心法? 用贪心法求解的问题需要满足以下特征:

(1) 最优子结构性质。当一个问题最优解包含其子问题的最优解时,称此问题具有最优子结构性质,也称此问题满足最优性原理。也就是说,从局部最优能扩展到全局最优。

① 一个简单的判断标准是,面值符合 $c_j > \sum_{i=1}^{j-1} c_i$ 的硬币,即任一面值的硬币,大于比它小的所有硬币的面值之和。可以用贪心法。例如以 2 的倍数递增的 1、2、4、8 等,这样的面值就符合条件。

面值是整数 c 的幂, c^0, c^1, \dots, c^k , 其中 $c > 1, k \geq 1$

• 99 •

(9) 143 页。把“最短距离”改为“最长距离”。

综上所述,距离结点 4 最远的距离用 dfs2()实现功能(2)。

状态的设计: 结点 i 的子树到 i 的往上走的最短距离 $dp[i][2]$ 。

```
#include <bits/stdc++.h>
using namespace std;
const int N = 10100;
```

(10) 157 页, 第 2 行, “用户” 改为 “读者”

第8章 数学

的乘积, 即 $a^{11} = a^{8+2+1} = a^8 \times a^2 \times a^1$ 。

要分别计算吗? 并不需要。用户可以容易地发现, $a^1 \times a^1 =$ 等, 都是 2 的倍数, 产生的 a^i 都是倍乘关系, 逐级递推就可以用 “base * = base;” 实现。

(11) 193 页。把 6013 改为 6103

【习题】

hdu 1062, 字符串反转。

hdu 6013, 字符串反转, 尺取法。

(12) 210 页。改为: 总复杂度是 $O(n(\log_2 n)^2)$

```
}
return 0;
}
```

上面的程序用到的 sort() 实际是快速排序, 每一步排 $\log_2 n$ 个步骤, 总复杂度是 $O(n \log_2 n)$ 。虽然已经很好——基数排序, 总复杂度只有 $O(n \log_2 n)$ 。在下一节的问和基数排序两种方案的倍增法程序, 执行时间分别是 1000

3. 基数排序

(13) 254 页。kruskal 算法, 把 “所有边都在 T 中”, 改为 “所有点都在 T 中”

kruskal 算法: 对边进行贪心操作。从最短的边开始, 把它加入到 T 中; 在剩下的边, 加入到 T 中; 继续这个过程, 直到所有点都在 T 中。

算法中, 重要的问题是判断圈。最小生成树显然不应该有圈, 否则就不是 “最小生成树”。在新加入一个点或者边的时候要同时判断是否形成了圈。

prim 算法

说明了 prim 算法的步骤。设最小生成树中的点的集合是 U , 开树为空, 所以 U 为空。



视频讲解

(14) 316 页。加一个视频：“第 12 章-表 12.1.mp4”。

合适的题目。大体上应该能考查竞赛队员的 5 种能力,包括编
知识、团队合作。难度上的区分如表 12.1 所示。

表 12.1 难度上的区分

奖牌	编码	计算思维	逻辑推理	算法知识
铜牌	**	**	**	**
银牌	***	***	***	***
金牌	****	****	****	****

从能力考查上看,铜牌 1 星或 2 星;银牌 3 星或 4 星;金牌
金牌的获奖比例按 3 : 2 : 1 逐渐缩小,可以得出结论:从铜牌: