普通高等教育“十一五”国家级规划教材 计算机系列教材

ACM/ICPC 算法基础训练教程

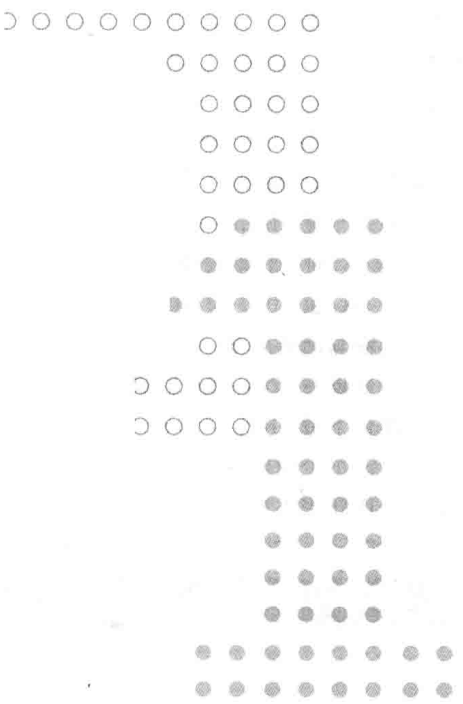
喻梅 于瑞国 主编

清华大学出版社

普通高等教育“十一五”国家级规划教材　计算机系列教材

喻梅　于瑞国　主编

ACM/ICPC 算法基础训练教程



清华大学出版社
北京

内 容 简 介

本书介绍 ACM/ICPC 的算法基础知识，主要内容包括基础算法、数据结构、搜索算法、图论基础、网络流(最大流、费用流、上下界网络流)、动态规划算法、数学基础、字符串算法以及计算几何基础。每一部分内容先介绍基本概念和基础理论，再通过例题讲解算法。书中所有例题均给出源程序代码及解题思路，便于读者学习和参考。

本书适用于刚刚步入 ACM/ICPC 的初学者，书中算法由浅入深，循序渐进，有利于初学者的学习。本书适合作为计算机及相关专业程序设计、数据结构和算法设计与分析等课程的教材，也可以作为计算机编程爱好者的参考书。

图书在版编目(CIP)数据

ACM/ICPC 算法基础训练教程/喻梅，于瑞国主编. —北京：清华大学出版社，2015
计算机系列教材
ISBN 978-7-302-41445-2

Ⅰ. ①A…　Ⅱ. ①喻…　②于…　Ⅲ. ①程序设计－算法－高等学校－教材　Ⅳ. ①TP311.5

中国版本图书馆 CIP 数据核字(2015)第 209639 号

责任编辑：张瑞庆
封面设计：常雪影
责任校对：梁　毅
责任印制：何　芊

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954
印 装 者：三河市少明印务有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　张**：26.25　　**字　　数**：654 千字
版　　次：2015 年 12 月第 1 版　　**印　　次**：2015 年 12 月第1 次印刷
印　　数：1～2000
定　　价：49.00 元

产品编号：064778-01

《ACM/ICPC 算法基础训练教程》 前 言

ACM 国际大学生程序设计竞赛(简称 ACM/ICPC)是世界上公认的规模最大、水平最高的国际大学生程序设计竞赛,一直受到国际各知名大学的重视,并受到世界各著名计算机公司的高度关注。ACM/ICPC 大赛的目的是让大学生充分展示自己运用计算机分析问题和解决问题的能力。在我国,越来越多的高校、越来越多的计算机及相关专业的学生关注并参与此项赛事,通过竞赛,培养学生牢固的专业素质,为后续专业的学习和研究打下良好的基础。

本书所介绍的内容均为 ACM/ICPC 基础算法,目的是使刚刚步入 ACM/ICPC、刚刚进入专业课学习的计算机专业学生以及程序设计爱好者了解计算机编程的基本算法。本书每一章知识点的介绍分为理论知识、例题解析、推荐学习三部分。通过对 ACM/ICPC 知识点基本概念和基础理论的介绍,了解相关的理论知识;通过对例题的解析,能够理解算法相应的知识点;通过习题的练习,掌握和巩固所学算法,以达到培养计算机专业人才的编程思想以及分析问题和解决问题的能力。本书注重基础算法的介绍,使初学者能够快速入门,并且对程序设计及算法产生兴趣,为后续进阶学习打下坚实的基础。

参与本书编写的人员均为天津大学 ACM/ICPC 代表队的现役及退役队员,在编写过程中参考了大量文献,结合多年的参赛经验,对本书的内容进行了撰写,并对书中例题的源程序代码进行了评测。

参与本书构思、撰写和审稿的人员有赵满坤、张敏杰、刘鑫、蒋星韬、吴建钢、毛洪玥、黄才宝、林榆旺、刘志强、阎杰、曹圣、刘凯、周挺乐、张家源。

在全书的撰写过程中,得到了清华大学出版社的大力支持,在此表示衷心的感谢。

由于时间仓促,作者水平有限,书中难免有不当之处,敬请读者批评指正。

编 者

2015 年 7 月

FOREWORD

第1章 基础算法

本章介绍最基础的算法，这些算法的思想简单明确，初学者应该首先熟练掌握这些算法。大部分情况下，这些算法不会作为单独的知识点进行考察，而是作为其他算法思想中的一部分，或者配合其他算法一起来求解问题。

1.1 模拟题

1.1.1 基本概念

模拟题就是按照题目给出的十分明确的规则对输入数据进行处理，并按照输出规则进行输出的题目。

模拟题有着非常鲜明的特征：

(1) 题目描述较长，主要是为了清晰地描述题目定义的规则(也有的题目因使用公认的规则而描述较短，如大数的四则运算等)。

(2) 基本不涉及高深的算法，也不需要对题目本身进行比较深入的思考。

(3) 代码量大，细节较多，出现错误不易排查。

模拟题有简单题也有复杂题。简单题规则较少，且浅显易懂，适合作为初学者的入门题目，用来熟悉编程语言的输入输出以及简单库函数的使用等；也有一些复杂的题目，规则很多，编程复杂度非常高，对代码的控制能力要求十分严格，少则百余行，多则数百行，稍不留意就会出现错误。

对于题目较长的模拟题，建议阅读题目的时候把关键的语句标注出来，以免编写代码的时候有疏忽遗漏的地方，当对某个细节感到模糊的时候，最好的方法是再次仔细阅读题目描述，而不是把自己的看法作为题目的规则，更不能妄加揣测、无中生有。

1.1.2 例题讲解

【例 1-1】 Counting Sheep。

题目描述：

共有 $n(n\leqslant 20)$组数据，其中每组数据有一个正整数 $m(m\leqslant 10)$，然后是 m 个单词，统计其中有多少个“sheep”。单词大小写敏感，因而“Sheep”等不匹配。

输入样例：

```
4
5
shep sheeps sheep ship Sheep
```

```
7
sheep sheep SHEEP sheep shepe shemp seep
10
sheep sheep sheep sheep sheep sheep sheep sheep sheep sheep
4
shape buffalo ram goat
```

输出样例：

```
Case 1: This list contains 1 sheep.
Case 2: This list contains 3 sheep.
Case 3: This list contains 10 sheep.
Case 4: This list contains 0 sheep.
```

题目来源：

TOJ 2001，East Central North America 2000 Practice

http://acm.tju.edu.cn/toj/showp2001.html

题目解析：

按照题目要求，总结规则如下：

(1) 计算每组数据包含多少个 sheep，注意大小写敏感。

(2) 相邻的两组输出之间包含一个空行。

参考代码：

```
#include<iostream>
#include<string>
using namespace std;

int main()
{
    int n, m, c=1, count;
    string str;
    cin>>n;
    while(n--) {
        if(c>1) cout<<endl;      //相邻的两组输出之间包含一个空行
        cin>>m;
        count=0;
        for(int i=0; i<m; i++) {
            cin>>str;
            if(str=="sheep") count++;
        }
        cout<<"Case "<<c++<<": This list contains "
        <<count<<" sheep."<<endl;
    }
```

```
    return 0;
}
```

【例 1-2】 VIM。

题目描述：

本题只有一组输入数据。第一行包含一个正整数 $L(L\leqslant 100)$，表示有一个包含 L 行文字的文本，接下来是 L 行文本，每行不超过 100 个字符。在文本之后是若干个(不超过 50 个)替换命令：[range]s/{pattern}/{string}/[flag]，命令中符号代表的含义如下：

- :表示替换命令的开始；
- [range]表示被操作的文本的范围；
- s 是 substitute 的简写，是替换的意思；
- {pattern}和{string}是要匹配的文本和将要替换成的文本；
- /用来作为分隔符；
- [flag]是一些操作的开关。

本题重要的规则和需要注意的事项说明如下：

(1) [range]一定会出现，为“%”(所有的行)或“a,b”(从第 a 行到第 b 行，$a<b$)。

(2) {pattern}和{string}只包含字母、数字、空格和下划线。

(3) 命令分割符包括/、~、!、@、#、$、%、^、&、*、()、-、+、=。

(4) [flag]只能是“g”，表示对出现的每个{pattern}都进行替换。

(5) 如果{pattern}为空，则使用上次不为空的{pattern}。

(6) 如果{pattern}是{string}的子串，不进行递归替换(这样永远替换不完)。

(7) 第 $i+1$ 条命令处理的文本是由第 i 条命令处理后的文本。

输入样例：

```
4
If the Tao is greet, then the operating system is greet.
If the operating system is greeter, then the compiler is greet.
If the compiler is greeter, then the applications is greet.
The user is pleased and there is harmony in the world.
:1,3s/greet/great/g
:%s//great/g
```

输出样例：

```
1  If the Tao is great, then the operating system is great.
2  If the operating system is greater, then the compiler is great.
3  If the compiler is greater, then the applications is great.
Pattern not found
```

题目来源：

HDU 3052，2009 Multi-University Training Contest 15-Host by BUAA

http://acm.hdu.edu.cn/showproblem.php?pid=3052