ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.7, No.2, January 2011,pp.359-361,369

E-mail: xsjl@cccc.net.cn
http://www.dnzs.net.cn
Tel:+86-551-5690963 5690964

不重复随机数的生成算法及应用

游兰菊

(枣庄学院,山东 枣庄 277160)

摘要:该文根据实际问题,提出了静态和动态生成不重复随机数的算法思想,详细剖析了这两种算法的基本原理,并给出了实现方法,解决了传统算法中普遍存在的低效和难以实现等问题。最后通过实例验证了这两种算法的有效性。

关键词:随机数;算法;不重复;过滤

中图分类号:TP312 **文献标识码**:A **文章编号**:1009-3044(2011)02-0359-03

The Generating Algorithm of Unrepeatable Random Numbers and its Implementation

YOU Lan-ju

(Administration Office, Zaozhuang University, Zaozhuang 277160, China)

Abstract: According to actual problem, this paper proposes an algorithm idea about static and dynamic algorithms to generate random numbers. By analyzing the basic principles of these two kinds of algorithms in detail, the paper shows two implementation methods, which solves the problems of poor efficiency and difficult implementation in traditional algorithm. In the end, two examples show the validity of these algorithms.

Key word: random number; algorithm; unrepeatable; filter

随机数在数据库应用、密码学、图像还原、数字模拟实验等方面有非常广泛的应用,目前,一般计算机语言都有生成随机数的函数,如 Delphi 语言中的 Random 函数、Visual Basic 语言中的 RND 函数等,但这些函数生成的是可重复的随机数列,若想获得不重复随机数列,则需要程序员自己编写代码来实现。目前虽然有不少文章在研究类似问题[1-2],但并没有彻底解决不重复随机数的生成问题;在专门研究不重复随机数生成及应用的文章[3-4]中,虽然给出了一些求解方案,但都存在重复试选、实际应用效果差等问题,不仅如此,这些文章研究的是条件不变情况下的随机数生成算法,然而,在很多实际应用中,不仅需要不重复的随机数,而且还需要动态条件下生成的随机数,如组卷时试题随机抽取问题,不仅要求随机抽出的题目不重复,而且要求每个试题的考核点也不相同,使得每次抽取试题时不仅要考虑已抽出试题本身对下次抽取的影响, 而且还要考虑所有与其考核点相同的试题对新抽取试题的影响。对这类条件不断变化下的随机数生成问题,使用固定条件下的随机数生成算法是难以解决的。本文将从解决此类实际问题出发,系统分析生成不重复随机数的算法原理,并给出快速高效的实现算法。

1 不重复随机数问题的提出

按生成条件的不同,不重复随机数有静态和动态方式,为便于说明问题,下面将这两类问题描述如下:

1)静态条件下不重复随机数生成问题

从取数范围 R 中随机抽取 M 个不重复的数 a_i (i=1,2,…,m),构成一个数列 $A=(a_1,a_2,\cdots,a_m)$,其中 $a_i \in R$ (i=1,2, …, m)。

2)动态条件下不重复随机数生成问题

动态条件是指生成新随机数时,不仅受已生成随机数的个体制约,而且还受与生成随机数相关的其它条件的制约,即每次生成随机数的条件是动态改变的。动态条件下不重复随机数的生成问题可描述如下:

从取数范围 R 中随机抽取 m 个不重复的数 a_i (i=1, 2, …, m) 构成一个数列 A,记为 $A=(a_1,a_2,\cdots,a_m)$,其中生成的第 i 个随机数 a_i 时不仅满足 $a_i \in R$,而且还必须满足条件 T_i,这里的 T_i 是与前 i-1 个随机数有关的动态条件。

下面将分析生成上述两种不重复随机数的算法。

2 随机数生成算法及实现

2.1 静态条件下不重复随机数生成算法

静态条件下生成不重复随机数的传统算法主要是试选法,基原理是利用随机函数,逐个生成指定范围内的随机数,当生成的一个随机数已经入选时,则放弃此数,再重新生成,直到找到 m 个不重复的随机数为止。此算法想法自然,实现容易,但试选可能造成死循环或长时间运算等问题,特别在较小取数范围内生成较多不重复随机数时,更可能出现这一问题。

试选法存在缺陷的原因是没有将已选数从被选数中过滤出去。若选中一个随机数后,立即将其从此后的被选数中删除,则肯定不会出现试选问题。下面算法就是基于这一思想,其原理是:将所有可选择的数存入一个数组,通过随机选择数组下标实现不重复随机数的选取,具体算法如下:

1)初始化数组 B,使 B 中保存范围 R 所有可选择的数;初始化数组 A,用于保存选出的随机数;初始化计数器 i,使 i=1。设 A 和

收稿日期:2010-12-06
基金项目:本文得到山东省教育厅科研项目基金资助(项目编号:J07WJ29)
作者简介:游兰菊(1963-),女,山东定陶人,中级职称,硕士,主要从事档案管理和计算机应用等方面的研究工作。

B 的下标都从 1 开始;n 为数组 B 中保存的可选数的个数;j 为临时整数型变量;

2)若 m>n,则“可选数不足”,转 7);

3)生成随机数 r,使 j = r,其中 r 表示取自于 1 至 n 之间的随机整数;

4)将 B 的第 j 个数组分量保存 A 到的第 i 个分量中,即 A (i)= B (j),并将 B 的第 n 个分量填充到第 i 分量中,使 B(i)=B(n),n=n1,即调整数组 B 中可选择数的个数 n,也就是取消数组 B 中已经选择的数;

5)i=i+1;如果 i<=m,则转 3);

6)则数组 A 中保存的就是所要的不重复随机数列;

7)程序结束。

由于上述算法每循环一次都去掉数组 B 的最后一个分量,因此称此方法为“去尾法”。通过分析此算法容易发现,除判断循环条件外,算法中全部是较简单的赋值语句,因此其算法复杂度是 O(n),可见算法达到了最佳性能,它可迅速生成满足用户条件的不重复随机数列。

另外,上述算法中并没有涉及随机数类型,只要将其保存到数组 B 中,就可生成不重复随机数列,因此,本算法生成的不重复随机数,并不特指通常使用的数,这是本算法的特色之一,下面实例可说明这一特征。

例 1:随机生成一个由 0~9,a~z,A~Z 不重复字符构成的 16 位序列号。

使用 Delphi 语言实现的编程代码如下:

```
procedure TForm1.Button1Click(Sender: TObject);
var A: array[1..16] of char;
B: array[1..62] of char;
j,i,n: integer;
begin
n:=62;          //置备选字符的个数为 62
for i:=1 to 10 do  //将 0~9 字符保存到数组 B 中
B[i]:=chr(47+i);
for i:=1 to 26 do  //将 A~Z 字符保存到数组 B 中
B[10+i]:=chr(64+i);
for i:=1 to 26 do  //将 a~z 字符保存到数组 B 中
B[36+i]:=chr(96+i);
Randomize;         //初始化随机数种子
for i:=1 to 16 do
begin
//取 1~ n 之间随机数用作 A 的下标
j:=round(random(n-1))+1;
A[i]:=B[j];    //保存 B 中第 j 个分量到 A[j]中
B[j]:=B[n];    //将 B 的第 n 个分量数填充到 B[j]中
n:=n-1;        //使可选字符总数减 1
end;
end;
```

运行上程序,则保存到数组 A 中的 16 个字符即为要求生成不重复随机数列。

2.2 动态条件下的不重复随机数生成算法

在生成随机数时,除考虑已经选出的随机数外,有时还要考虑与选出随机数相关的其它条件,此种选取随机数的算法仍使用“选出则过滤”思想,即每选取出一个新随机数后,除将该是随机数过滤出去外,还要过滤所有满足动态条件的可选数,然后再选取下一个随机数。重复上过程,直到选择出所有需求的随机数为止。具体算法如下:

1)初始化数组 B,使 B 保存范围 R 所有可选择的数;初始化数组 A,用于保存选出的随机数;初始化随机数个数计数器 i,使 i=1。设 A 和 B 的下标都从 1 开始;n 用于保存 B 中存放的可选数总数; j 为临时整数型变量;

2)如果 m > n,则“可选数不足”,转 8);

3)生成随机数 r,使 j = r,其中 r 表示自 1 至 n 范围内生成的随机整数;

4)将 B 的第 j 个分量保存 A 到的第 i 个分量中,即 A (i)= B (j);

5)将 B 中所有与 B (j)相关且满足给定条件?Tj 的可选数过滤出去,设共过滤出 F_j 个数据,则将数组 B 前 $n-F_j$ 分量中过滤出去的分量,由 B 中从第 $n-F_j+1$ 分量到第 n 个分量中没有过滤出的可选数填充,则数组 B 中 1 到$(N-F_j)$个分量全部为符合条件的可选数,使 $n = n-F_j$;

6)若 n < m-i ,则表示可选数不足,转 8);

7)i=i+1;如果 i <= m,则转 3);

8)程序结束。

算法结束后,在数组 A 中保存的就是所求的不重复随机数列

从上述程序可以看出,动态算法的原理是静态条件下“去尾”算法的扩展,只是静态算法每次从数组中去掉一个已抽取的数,而动态算法一次可能要过滤掉多个可选数,若把抽出的随机数看成自相关的话,则静态算法只是动态算法的一个特例。

对多个分量的过滤与填充,可采用“先标注、后填充、再去尾”的方法,此方法不仅运算速度快,而且易于实现,具体原理见图 1,

设数组 B 有 9 个分量,分别存放从 i 到 q 的 9 个字符。若第一次在 1~9 之间生成的随机数为 3,即 i = 3,则第一个选择出的随机数为 B[3],即“k”。假设与此字符“k”相关还有 B 的第 5、8 个分量,则先将他们三个分别标注为-1(见图 1);再将数组 B 中前 6(即 9-3)个分量中值为-1 的分量分别由数组 B 最后 3 个没有标注-1 的分量填充,则 B 重新构成一个有 6 个分量的新数组,再重复选取过程,即可完成随机数的选取工作。

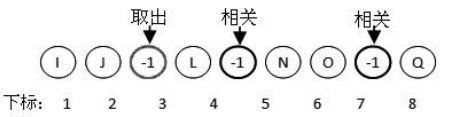

图 1 数组 B 标注后填充前的图示

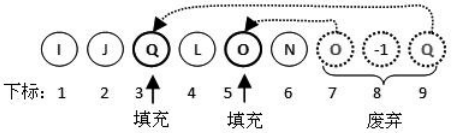

图 2 填充后数组 B 的图示。B 的后 3 个分量将废弃

例 2,某高校要求从具有 20 个考核点 100 道试题的试题库中随机抽取 10 道有不同考核点的试题组成试卷,其中考核点由 1~20 个数字表示,题目编号及其对应的考核点如表 1。

表 1

题号	1	2	3	4	5	6	7	8	9	10
考核点	3	7	8	13	9	1	14	2	19	2
题号	11	12	13	14	15	16	17	18	19	20
考核点	15	1	20	13	3	10	9	11	19	4
题号	21	22	23	24	25	26	27	28	29	30
考核点	2	8	6	13	19	18	14	20	9	1
题号	31	32	33	34	35	36	37	38	39	40
考核点	14	15	1	14	15	13	10	1	14	3
题号	41	42	43	44	45	46	47	48	49	50
考核点	20	15	14	8	7	9	9	20	5	6
题号	51	52	53	54	55	56	57	58	59	60
考核点	13	11	10	7	14	2	7	10	11	12
题号	61	62	63	64	65	66	67	68	69	70
考核点	5	20	6	14	5	7	8	7	7	1
题号	71	72	73	74	75	76	77	78	79	80
考核点	13	7	12	8	7	18	18	5	15	19
题号	81	82	83	84	85	86	87	88	89	90
考核点	16	18	11	5	11	10	9	17	4	8
题号	91	92	93	94	95	96	97	98	99	100
考核点	16	5	2	16	20	13	3	6	19	5

为方便编程,假设数组 B 中保存试题的编号,数组 C 保存与数组 B 各分量代表试题对应考核点(如 C[2]=10 表示 B[2]对应试题的考核点为 10),抽出的试题编号保存到数组 A 中。

算法实现代码如下:

```
procedure TForm1.Button1Click(Sender: TObject);
var
B,C: array[1..100] of integer;
A: array[1..10] of integer;
len,j,i,k,s,r,n,tnum,id: integer;
begin
Randomize;
n:=100;
for i:=1 to n do
begin
B[i]:=i;    //保存试题的题号
//生成对应考核点模拟数据到 C 中
C[i]:=random(20)+1; //考核点编号不大于 20
end ;
for i:=1 to 10 do  begin
j:=round(random(n-1))+1;
id:=C[j];    //保存考核点数据
A[i]:=B[j];  //保存抽取的数到数组 A 中
k:=0;
for r:=1 to n do begin
if C[r]=id then begin //判断有否相同考核点
B[r]:=-1;        //设置过滤标记
k:=k+1;         //累计被过滤的试题个数
end;
end;
n:=n-k;        //计算剩余可选试题个数
//填充数组 A 的前 n 分量中值为-1 的分量
k:=1; s:=n ;
for r:=1 to n do
begin
if a[k]<>-1 then  inc(k); //搜索被填充的分量
if a[s]=-1  then dec(s);  //搜索填充分量
if k=s    then  break;  //判断是否搜索完成
if (a[k]=-1) and (a[s]<>-1) then
a[k]:=a[s];       //填补分量
end;
end
end;
```

运行上述程序,则在 A 中可得到的一组满足选择条件的试题编号,形如:A = (31, 98, 76, 60,20, 96, 22, 78, 15, 58)。在实际应用中,B、C 数组中的值可通过编程从试题库相应的字段中获取,得到 A 后,按照 A 中保存的题号从试题库中抽出相应试题即可完成组卷。

3 结束语

本文给出的两类随机数生成算法具有广泛的应用性,基本涵盖不重复随机数的各种应用。由于动态条件的不确定性,算法中没有给出动态条件的具体规范,但每次抽出随机数后,及时将满足过滤条件的可选数过滤出去是解决问题的关键,例 2 正是按这一思想解决了动态条件下不重复随机数生成算法,该例具有普适性,只要稍加修改,就能将实际应用中各种动态条件化归为此例求解方法,此算法在作者的硕士论文[5]中得到较好的运用。

(下转第 369 页)

3 网上购物的实现

3.1 系统功能描述

商务网站一般情况下是由前台系统和后台系统两部分构成的。前台系统是普通用户浏览网站页面,在那里可以进行的一般操作如:注册、会员登录、浏览商品、购物、留言、支付等。后台系统一是对商品的信息进行管理、发布和修改;二是要进行会员管理、商品配送、账务管理、报表统计;三是要保障系统运转及账号的安全管理。

3.2 开发工具及平台选择

ASP 服务器端执行脚本指令环境是目前较先进的方法,数据库可以选用 SQL Server。这样选便于系统的改进和扩充。这也是目前众多网站建设所使用的开发工具。

3.3 数据库设计

1)商品基本信息表:该表主要用于记录商品的基本信息(包括商品的名称、价格、型号、产地、分类属性、商品详细信息、图片、供货商情况等)。

2)会员资料表:记录会员的注册、登录、操作记录、客户等资料。

3)购物车表:用于记录每次购物的记录并具有统计功能。

4)订单表:每个会员有一个订单表,记录其消费情况等基本信息。

5)配送记录:各种付款送货信息。

3.4 功能模块设计

3.4.1 会员管理模块

用于实现会员注册,会员登陆,权限功能检查,留言等功能。权限功能检查是指登录人员登录之后,系统自动通过登录人员的身份和口令,将其具备的功能权限转化成二进制的码串,以备后续程序调用。

3.4.2 购物页面

主要用于向用户展示用户希望了解的商品。根据不同需求可以对商品进行查询,如按品牌查询、按关键字查询、按价格排序等。

3.4.3 购物篮

主要实现两个功能。1)增加商品到购物车:若会员找到购买的商品,点击购买后将所选商品的信息如:商品编码、商品名称、销售价、购买价、购买数量记入购物车表中。2)显示购物车信息:点击购物车后显示相应的商品信息:商品编码、商品名称、销售价等。

3.4.4 付款

进入付款页面,调出会员记录,显示收件人和付款人资料,选择付款方式,并将以上信息写入订单表中。

3.4.5 后台数据管理

1)商品管理:实现系统管理员对商品的管理。如查看商品目录、增删商品种类、删除已不存在的商品或修改、增加商品信息。

2)会员管理:具体实现对网站会员账户的各种管理,包括新增会员及权限分配。给会员发送信息、查看会员资料、更新会员资料和删除不合法会员等。

3)订单管理:提供及时有效的订单查询检索。具体实现:根据订单发货情况、更改订单处理情况、检查订单情况、订单转储和查看所有订单。

4)财务管理:有完善的会计科目体系和灵活的会计科目重定义功能,严谨的会计审核处理系统,严格按国家财会制度进行财务核算、制单和输出报表。

3.4.6 系统维护模块

系统对会员信息、系统信息、以及各种单据信息进行维护和管理,比如审核批准会员、删除一定时间范围作废的单据、转储数据库等,以保障系统的正常运行。

相信随着软件工程的发展和进一步成熟,商务网站建设的发展会取得更大的进步。在实际项目中,我们要坚持发送软件工程的管理,并在实践中总结适合自身的经验,这样才有利于管理技术的进步和软件项目的顺利完成,创造出更高的品质、更大的效益。

参考文献:

[1] 刘欣怡,周跃东,田秀丽.软件工程[M].北京:清华大学出版社,2007.

[2] 周树清.电子商务情景案例[M].北京:中国国际广播出版社,2001.

[3] 姚国章.中国企业电子商务发展战略[M].北京:北京大学出版社,2001.

(上接第 361 页)

参考文献:

[1] 尹柯.随机选题算法的设计与实现[J].河南大学学报:自然科学版,2004,34(1):91-93.

[2] 陈溪辉.完全随机数的生成及应用[J].衡阳师范学院学报:自然科学,2003,2(l2):30-32

[3] 卢守东.PowerBuilder 中随机功能的设计与实现[J].福建电脑,2005(1):66-67.

[4] 张仿.随机数在加密技术中的应用分析[J].计算机应用与软件,2004,21(12):105-107.

[5] 游兰菊.考试及成绩管理过程中的防协同作弊技术及其应用[D].上海:华东师范大学硕士毕业论文,2008.