



# 随机抽取试题的四种算法

## ——网上考试系统编制心得

文/ 谢作如

因为教学的需要,笔者编写过一个ASP+MS SQL2000的网上考试系统,其功能主要为:实现判断题、单项、多项选择题和填空题的在线自动答题、改卷;并将学生的错误答案记入数据库,供教师分析。在编写从题库中随机抽取试题这一模块的算法上,颇费了一番周折,现将解决过程记录如下,以供大家参考。

为了便于说明问题,文中提供的代码中的变量pd为从题库中要抽取出来考试的试题数量,数据库表名与字段名都使用了中文,并仅以判断题为例。

### 算法一

其思路为先将数据库中所有数据读出,获得试题的总数后,生成一个1~(试题的总数-考试的试题数量)之间的随机数,然后从这里开始读出数据:

```
<% set rs=server.CreateObject("ADODB.
RecordSet")

sql="select * from 判断题 order by id asc"
rs.open sql,conn,1,1

mycount=rs.Recordcount '取得试题总数
randomize '初始化随机数种子值
n=fix((mycount-pd+1)*Rnd+1)
rs.move n '指针移到n这个随机数位置
for i=1 to pd
session("pdda")=session("pdda")&rs("正确答案")
&"|" '用session来记录标准答案
'输出试题及答案%>
```

```
<tr>
  |
```

这种算法基本上可以实现随机抽取试题,并让每个学生的试题和每一次刷新以后的试题都不相同,但是它的最大不足在于试题的先后顺序总是相同,特别是题库中试题不多的时候,学生几乎可以用背答案的方法来应付考试。虽然可以通过改变数据的排序方式来改变试题的先后顺序,但变化总是不大。

### 算法二

第二种算法的思路很简单,就是不断生成1~(题库中的试题总数)之间的随机数,然后到数据库中读取这条记录,直到满足考试的试题量为止。

```
< %
set rs=server.CreateObject("ADODB.RecordSet")
sql="select * from 判断题 order by id asc"
rs.open sql,conn,1,1
```

```

mycount=rs.Recordcount '取得题库中的试题
总数
rs.close
for i=1 to pd
randomize
sid=int((mycount +1)*rnd+1) '生成1~题库
中的试题总数之间的随机数
set rs=conn.execute("select * from判断题where
id=&sid)
while rs.eof
randomize
sid=int((mycount +1)*rnd+1)
set rs=conn.execute("select * from判断题where
id=&sid) '如果数据库中找不到这条试题，就继续
生成随机数读取试题
wend
session("pdda")=session("pdda")&rs("正确答案")
&"|" '用session来记录标准答案
'输出试题及答案%>
<tr>
<td width="10%" ><%=i%>、<%=rs("题
目内容")%></td>
<td align="center" width="10%" ><select
name="cate<%=i%>">
<option selected value=True>对</option>
<option value=False>错</option></select>
</td>
</tr>
<%
next
%>

```

这种算法应该是真正意义上的随机抽取试题，但是遗憾的是如果在题库中题量不多的情况下，很容易会在数据库中读取重复的试题，虽然也可以再使用一个变量或数组来存储已经读取过的试题id来解决试题重复的问题，但算法过于繁琐，笔者认为并不可取。

### 算法三

由于第二种算法容易造成试题重复，为了避免系统产生重复的随机数，试着将题库中试题总数均分为kp个范围，让每个范围产生一个随机数，这样就有效地避免了随机数的重复。

```

<% set rs=server.CreateObject("ADODB.
RecordSet")

sql="select * from 判断题 order by id asc"
rs.open sql,conn,1,1
mycount=rs.Recordcount '取得试题总数
for i=1 to pd
randomize
temp=fix((fix(rs.Recordcount/pd)+1)
*rnd+1) '生成1~题库试题总数除以试卷试题数
之间的随机数
rs.move temp '指针移到随机数位置
session("pdda")=session("pdda")&rs("正确答案")
&"|" '用session来记录标准答案
'输出试题及答案%>
<tr>
<td width="10%" ><%=i%>、<%=rs("题
目内容")%></td>
<td align="center" width="10%" ><select
name="cate<%=i%>">
<option selected value=True>对</option>
<option value=False>错</option></select>
</td>
</tr>
<%next
rs.close%>

```

这种算法能够有效地解决算法一和算法二的不足，既做到了随机抽取试题，又做到了试题不重复。但是仔细一想还是存在不足：就是题库中的每一道试题出现的概率不相同，这样就显得不科学了。因为kp次都产生大数字的概率不大，这样排在后面的试题出现的机会很小。

## 算法四

算法四是笔者最后的研究结果，其算法分为三步：

Step1 获取试题库试题总数，然后生成一个1~试题总数的阵列。

Step2 生成随机数，将这个矩阵打乱。

Step3 按顺序取出阵列中的题目。

这种算法和洗牌的原理相类似。

(设试题库总数为10，要抽取5道题)

Step1：阵列的初始内容如表1所示。

Step2：生成两个随机数，如3和6。然后将A3和A6的内容交换，阵列的内容变为如表2所示。

Step3：按顺序取出阵列中的题目A1~A5的内容，应该是1、2、6、4、5，读出数据库中相应的试题。

如果不断循环Step2，该阵列中的内容就随机打乱，这样既实现了随机抽取试题的目的，又避免了试题重复抽取。

```
< %
dim matrix() '定义变量数组
set rs=server.CreateObject("ADODB.RecordSet")
sql="select * from 判断题order by id asc"
rs.open sql,conn,1,1
mycount=rs.Recordcount '取得试题总数
'设定阵列的初始值
for i=0 to mycount
matrix(i)=i+1
next
randomize '生成随机数种子
for i=0 to 2*mycount '循环2*试题总次数
```

表1

A 1	A 2	A 3	A 4	A 5	A 6	A 7	A 8	A 9	A 10
1	2	3	4	5	6	7	8	9	10

表2

A 1	A 2	A 3	A 4	A 5	A 6	A 7	A 8	A 9	A 10
1	2	6	4	5	3	7	8	9	10

```
j=fix(rnd*mycount)
k=fix(rnd*mycount)
'交换matrix(k)和matrix(j)的内容
temp=matrix(k)
matrix(k)=matrix(j)
matrix(j)=temp
next
'取出阵列中的题目，数量为试卷中该类题的数量
for i=1 to pd
rs.absolutePosition=matrix(i) '把记录指针移指向第matrix(i)条记录
session("pdda")=session("pdda")&rs("正确答案")&"|" '用session来记录标准答案
'输出试题及答案%>
<tr>
<td width="10%" ><%=i%>、<%=rs("题目内容")%></td>
<td align="center" width="10%" ><select name="cate"><%=i%>>
<option selected value=True>对</option>
<option value=False>错</option></select>
</td>
</tr>
<%next
rs.close%>
```

以上代码在Windows 2000 Server +SQL 2000、Windows 2000 Server+Access调试通过。程序演示地址：[Http://www.kpcn.org/test/](http://www.kpcn.org/test/)。

(作者单位：浙江省平阳县职业教育中心 325400)