

## ,10, 概率整章注解

2018年4月6日 15:34

概率的应用：

- 1, 利用随机来改进著名算法, 快排
- 2, 随机数发生器, 用给定的随机数发生器构造另一个

问题：某地区重男轻女，一个家庭如果生出一个女孩就一直生，直到生出男孩就停止生育，假设一胎只会生出一个孩子，问时间足够长之后，男女比例会变为多少。

答：

每个孩子出生男女概率是50%，所以每次出生的男女比例是相同的。

假设这个国家有 $n$ 对夫妇，那么 $n$ 对夫妇将生下 $n$ 个男孩，这 $n$ 个男孩是这样生下的，假设生男生女的概率是50%，那么 $n/2$ 个男孩是第一胎生下的，同时将有 $n/2$ 个女孩生下， $n/2$ 对生女孩的夫妇将继续生，其中 $n/4$ 的夫妇生下男孩， $n/4$ 的夫妇继续生下女孩，然后是 $n/8$ 的夫妇顺利得到男孩，又有 $n/8$ 的夫妇生下女孩，依此类推，这个国家将生下 $n/2 + n/4 + n/8 + \dots$ 的女孩，所以男女比例是 $n : (n/2 + n/4 + n/8 + \dots) = n : n = 1 : 1$

来自 <[https://blog.csdn.net/java2000\\_net/article/details/3118698](https://blog.csdn.net/java2000_net/article/details/3118698)>

问题：有一个机器按自然数序列的方式吐出球，1号球，2号球，3号球等等，你有一个袋子，袋子里最多能装下 $k$ 个球，并且除了袋子外，你没有更多的空间，一个球一旦扔掉，就再也不可拿回。设计一种选择方式，使得当机器吐出 $N$ 号球时，你袋子里的球数是 $k$ 个，同时可以保证从1号球到 $N$ 号球的每一个，被选进袋子的概率都是 $k/n$ 。

这是一个蓄水池抽样问题，算法过程如下

- 1, 处理1- $k$ 号球时，直接放入袋子
- 2, 处理第 $i$ 号球 ( $i > k$ )，以 $k/i$ 的概率决定是否将其放入。若放，则在袋子中随机丢弃一个，放入 $i$ 。否则，直接扔掉
- 3, 处理第 $i+1$ 号球，重复步骤1or2.

证明过程：

假设第 $i$ 号球被选中，那么在选第 $k+1$ 号球之前，第 $i$ 号球留在袋子里的概率为1。

在选第 $k+1$ 号球时，什么情况下 $i$ 会被淘汰？只有决定将 $k+1$ 号球放进袋子，同时袋子里的 $i$ 号球被随机选中并丢掉，也就是这两种情况同时发生，才会被淘汰，概率为  $k/k+1 * (1/k) = 1/k+1$ 。

那么选 $k+1$ 号球时， $i$ 号球留下的概率为  $1 - (1/k+1) = k/k+1$

在选 $k+2$ 号球时，同样的， $i$ 被淘汰的概率为  $k/k+2 * (1/k) = 1/(k+2)$ 。留下的概率为  $k+1/k+2$

那么从1号球到第 $k+2$ 号球， $i$ 号留下的概率为 选 $k+1$ 和选 $k+2$ 时的概率之积， $k/k+1 * k+1/k+2 = k/k+2$

以此类推，在选第 $n$ 号球时， $i$ 号球留在袋子里的概率为 $k/n$ 。

1,



下一题

n只蚂蚁从正n边形的n个定点沿着边移动，速度是相同的，问它们碰头的概率是多少？  
给定一个正整数n，请返回一个数组，其中两个元素分别为结果的分子和分母，请化为最简分数。  
测试样例：

3

返回: [3,4]

C/C++ (clang++ 3.3)

```
1 class Ants {
2 public:
3     vector<int> collision(int n) {
4         // write code here
5     }
6 };
```

屏幕剪辑的捕获时间: 2018/4/6 15:35

/\*

问题：三只蚂蚁从正三角形的三个顶点沿着边移动，速度是相同的，问它们碰头的概率多大  
每只蚂蚁有两种选择，顺指针，逆时针。

因此，一共有 $2^3 = 8$ 种，只有都顺or都逆才不会相遇

n只蚂蚁从正n边形的n个顶点沿着边移动，求碰头概率

给定一个整数n，返回一个数组，其中两个元素分别为分子，分母，请化为最简分数

\*/

#include<bits/stdc++.h>

using namespace std;

class Solution

{

public:

vector<int> calc(int n)

{

vector<int> res;

int up = 0;

int down = pow(2,n);//分子分母

up = down - 2;

int comm = gcd(up, down);

res.push\_back(up/comm);

res.push\_back(down/comm);

return res;

}

int gcd(int x, int y)//计算最大公约数

if(y == 0)

return x;

else

return gcd(y, x%y);

}

};

int main()

{

int n =3;

Solution s;

vector<int> res = s.calc(n);

vector<int>::iterator iter;


for(iter = res.begin(); iter!=res.end(); iter++)

cout<<\*iter<<" ";

来自 <<http://tool.oschina.net/highlight>>

直通BAT面试算法精讲课 > 概率 > 10.5 随机01练习题



```
C/C++ (clang++ 3.3)   
1 class RandomP {  
2 public:  
3     static int f();  
4 };  
5  
6 class Random01 {  
7 public:  
8     // 用RandomP::f()实现等概率的01返回  
9     int random01()  
10    {  
11        int a,b;  
12        while(1)  
13        {  
14            a=RandomP::f();  
15            b=RandomP::f();  
16            if(a!=b)  
17                return (a>b)?0:1;  
18        }  
19    }
```

```
#include<bits/stdc++.h>
using namespace std;
class Solution
{
public:
    int random_p()//以p的概率生成0, 1-p的概率生成1
    {
        int r = rand()%10+1;//产生1-10的随机数
        if(r>6)
            return 1;//1的概率2/5
        else
            return 0;//0的概率3/5
    }

    int random_01()//用random_p()函数生成等概率生成0, 1的随机函数
    {
        while(1)
        {
            int a = random_p();
            int b = random_p();//连续调用两次
            if(a!=b)//01 or 10的情况
            {
                return a>b?0:1;//分别返回0, 1; 终止while, 否则相等, 则继续产生, 直到产生01, or10.
            }
        }
    }
};
```

```

    }
}
};
int main()
{
    Solution s;
    for(int i = 0; i < 50; i++)
        cout<<s.random_01()<<" ";
    cout<<endl;
    return 0;
}

```

来自 <<http://tool.oschina.net/highlight>>

,3,

直通BAT面试算法精讲 > 概率 > 10.4 随机函数练习题

足球比赛练习题

给定一个等概率随机产生1~5的随机函数，除此之外，不能使用任何额外的随机机制，请实现等概率随机产生1~7的随机函数。(给定一个可调用的Random5::random()方法,可以等概率地随机产生1~5的随机函数)

```

C/C++ (clang++ 3.3)
1 // 以下内容请不要修改
2 class Random5 {
3 public:
4     static int randomNumber();
5 };
6
7 class Random7 {
8 public:
9     int rand5() {
10         return Random5::randomNumber();
11     }
12     // 以上内容请不要修改
13
14
15     int randomNumber()
16     {
17         int a=5*(rand5()-1)+rand5()-1;

```

屏幕剪辑的捕获时间: 2018/4/6 15:35

\*问题：给定一个等概率随机产生1~5的随机函数，除此之外，不能使用任何额外的随机机制，请实现等概率产生1~7的随机函数

给出了一个可调用的Random5::random()方法，可以等概率生成1-5

解题思路：

1~5无法表示1~7，所以我们需要将1~5扩展，形成更大的长度以便容得下1~7

1. R() 函数可以产生 1~5

2. R()-1 函数可以产生0~4

3. ( R()-1 ) \*5函数可以产生0、5、10、15、20

4. ( R()-1 ) \*5+R()-1 函数便可以产生0~24，

这个过程，生成的数如果大于20，则重新生成，直到获取到的数是0~20以内的，那么就相当于把21~24这部分多出来的概率分摊到了0-20

5. 得到了可以容纳0~20的表达式，然后对其进行模7，0-6, 7-13, 14-20 模7结果一样，可以产生0~6的表达式

6. 最后即对上述进行加1即可。

C/C++ 中rand()和random()函数

```

int rand(void);
// 返回一个随机数0 ~ pow(2, sizeof(int))-1
long int random(void);

```

返回一个随机数  $0 \sim \text{pow}(2, \text{sizeof}(\text{long int})) - 1$

用下列公式即可得到指定范围  $[m, n]$  的随机数:

```
r = rand()%(n - m + 1) + m;
```

```
*/
#include<bits/stdc++.h>
using namespace std;
class Random5
{
public:
    static int random5()
    {
        return rand()%5+1;
    }
};

class Random7
{
public:
    int random5()
    {
        return Random5::random5();
    }

    int random7()
    {
        int num = (random5()-1)*5 + random5()-1;
        while(num>20)
            num = (random5()-1)*5 + random5()-1;
        num = num % 7 + 1;
        return num;
    }
};

int main()
{
    Random7 r7;

    for(int i = 0; i < 50; i++)
    {
        cout<<r7.random7()<<" ";
    }

    cout<<endl;

    return 0;
}
```

来自 <<http://tool.oschina.net/highlight>>

,4 ,

假设函数 $f()$ 等概率随机返回一个在 $[0,1)$ 范围上的浮点数，那么我们知道，在 $[0,x)$ 区间上的数出现的概率为 $x(0 < x \leq 1)$ 。给定一个大于0的整数 $k$ ，并且可以使用 $f()$ 函数，请实现一个函数依然返回在 $[0,1)$ 范围上的数，但是在 $[0,x)$ 区间上的数出现的概率为 $x$ 的 $k$ 次方。

```
C/C++ (clang++ 3.3)
1 class RandomSeg {
2 public:
3     // 等概率返回[0,1)
4     double f() {
5         return rand() * 1.0 / RAND_MAX;
6     }
7     // 通过调用f()来实现
8     double random(int k, double x)
9     {
10         double res=-1;
11         for(int i=0;i<k;i++)
12         {
```

屏幕剪辑的捕获时间: 2018/4/6 15:35:5,

/\*

问题：假设函数 $f()$ 等概率的返回一个在 $[0,1)$ 之间的浮点数。那么我们知道，在 $[0,x)$ 区间上的数出现的概率为 $x(0 < x \leq 1)$ 。给定一个大于0的整数 $k$ ，并且可以使用 $f()$ 函数，请实现一个函数依然返回在 $[0,1)$ 范围的数，但在 $[0,x)$ 区间上的数出现的概率为 $x$ 的 $k$ 次方。

先看如何得到 $x$ 的二次方，很简单，调用两次 $f()$ ，取较大的那个数，如果较大的落在 $[0,x)$ ，那么较小的那个也肯定是。同理，调用 $k$ 次，取最大的数仍小于 $x$ 即可。

\*/

#include<bits/stdc++.h>

using namespace std;

class Solution

{

public:

```
double random_01()//等概率返回[0,1)上的浮点数，出现在[0,x)上的概率为x，注意这里返回的是都崩了类型
{
    return rand() * (1.0 / RAND_MAX);
}
```

```
double rand_01(int k)//返回[0,1]上的浮点数，出现在[0,x]上的概率为 $x^k$ ;
```

```
{
    double res = -1;
```

```
    for(int i = 0; i < k; i++)
```

```
        double d = random_01();
        res = max(res, d);
```

```
    }
```

```
    return res;
```

```
}
```

```
};
```

```
int main()
```

```
{
```

```
    Solution s;
```

```
    for(int i = 0; i < 50; i++)
```

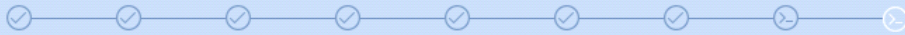
```
        cout<<s.rand_01(5)<<" ";
```

```
    cout<<endl;
```

```
    return 0;
```

```
}
```

来自 <<http://tool.oschina.net/highlight>>



下一题

给定一个长度为N且没有重复元素的数组arr和一个整数M，实现函数等概率随机打印arr中的M个数。

```
C/C++ (clang++ 3.3)
1 class RandomPrint {
2 public:
3     vector<int> print(vector<int> arr, int N, int M)
4     {
5         vector<int> res;
6         for(int i=0;i<M;i++)
7         {
8             int pos=rand()%(N-i);
9             res.push_back(arr[pos]);
10            swap(arr[pos],arr[N-i-1]);
11        }
12        return res;
13    }
14 }
```

屏幕剪辑的捕获时间: 2018/4/6 15:36

/\*问题: 给定一个长度为N且没有重复元素的数组arr和一个整数M，实现函数等概率的打印arr中的M个数。

这里有个很好的等概率打印小技巧，首先从0~n-1中随机定位一个位置a，打印，并将其与n-1位置的数交换，然后从0~n-2里面做同样操作，直到打印m个数。

```
*/
#include<bits/stdc++.h>
using namespace std;
class Solution
{
public:
    void solution(vector<int> arr,int n,int m)
    {
        for(int i = 0;i<m;i++)//执行m次
        {
            int pos = rand()%(n-i);//rand()%n,生成0~n-1的随机数。
            cout<<arr[pos]<< " ";
            int temp = arr[n-1-i];
            arr[n-1-i] = arr[pos];
            arr[pos] = temp;
        }
        cout<<endl;
    }
};

int main()
{
    int n = 7;
    int m = 4;
    int a[n] = {1,2,4,5,6,7,9};
    vector<int> arr(a,a+n);
    Solution s;
    s.solution(arr,n,m);
    return 0;
}
```

来自 <<http://tool.oschina.net/highlight>>

,6 ,

下一题

有 $2k$ 只球队，有 $k-1$ 个强队，其余都是弱队，随机把它们分成 $k$ 组比赛，每组两个队，问两强相遇的概率是多大？  
 给定一个数 $k$ ，请返回一个数组，其中有两个元素，分别为最终结果的分子和分母，请化成最简分数  
 测试样例：

4

返回: [3, 7]

C/C++ (clang++ 3.3)

```
1 class Championship {
2 public:
3     vector<int> calc(int k)
4     {
5         vector<int> res;
6         int up=1,down=1,i=2*k-1;
7         while(i>0)
8         {
9             down*=i;
10            i=i-1;
11        }
12        while(i>0)
13        {
14            up*=i;
15            i=i-1;
16        }
17        return res;
18    }
19 }
```

屏幕剪辑的捕获时间: 2018/4/6 15:36

/\*问题：8只球队，有3个强队，其余都是弱队，随机把他们分成四组比赛  
 每组两个队，问两强相遇的概率。

首先计算8支球队组成4队的方法数：

首先任选一只队伍，有7种选择对手的方式，

再选一只队伍，有5种选择对手的方式

再选一只，有三种

最后一支，一种

共有 $7*5*3*1 = 105$ 种

再计算没有两强相遇的概率：

从5支弱队选中三只与强者配对，剩下两只自行配对

$c(5, 3)*A(3, 3) = 60$  种

$A(3, 3)$ 为三只强三只弱队彼此配对的方法数。

故两强相遇的概率为 $1 - 60/105$ ;

问题：有 $2k$ 只球队， $k-1$ 个强队，其余都是弱队，随机把他们分成 $k$ 组比赛，问两强相遇的概率。

给定一个数 $k$ ，请返回一个数组，其中有两个元素，分别为最终分子分母，请化简成最简分数。

```
*/
#include<bits/stdc++.h>
using namespace std;
class Solution
{
public:
    vector<int> calc(int k)//得到化简后的分子分母数组
    {
        vector<int> res;
        int up = 1;//这里初始化一定要为1，因为这里是累乘
        int down = 1;//分子分母，分别代表两强相遇的情况数，和所有组队情况数
        for(int i = 1;i<=k;i++)//组的第几队
        {
            down*=2*(k-i)+1;//根据上面找的规律公式
        }

        for(int i = k+1;i>0;i--)
        {
            up*=i;//根据上面分析找规律公式，用for循环计算出来即可
        }
        up = down - up/2;
        int comm = gcd(up, down);
        res.push_back(up/comm);
        res.push_back(down/comm);
        return res;
    }
    int gcd(int x, int y)//得到最大公约数
```



```

    {
        if (y==0)
            return x;
        else
        {
            return gcd(y, x%y); //这种情况，一旦找到最大公约数递归即可停止。
        }
    }
};

int main()
{
    int k = 4;
    Solution s;

    vector<int> res = s.calc(k);
    vector<int>::iterator iter;
    for(iter = res.begin(); iter!=res.end(); iter++)
        cout<<*iter<<" ";
    cout<<endl;
    //cout<<s.gcd(3, 21)<<endl;
    return 0;
}

```

来自 <<http://tool.oschina.net/highlight>>