



## 2015 多易网络技术岗

一、有 20 张上下表面光滑的扑克牌，其中有 8 张向上，要求你闭着眼睛且不借助任何工具把这 20 张扑克牌分成两堆，使得每堆向上的扑克牌的数目一样多

答案：首先，把扑克牌随机分成两堆，一堆 12 张，一堆 8 张，假设 12 张的里面有  $X$  张朝上的，那么 8 张的里面就有  $8-X$  张朝上的， $X$  张朝下的，此时，把 8 张扑克牌全部反面，那么朝上的也变成了  $X$  张。

二、有一串玛瑙项链，项链上面有  $N$  个玛瑙珠子，这些玛瑙有  $M$  ( $M \leq 9$ ) 种不同的颜色，购买的时候可以只选择购买项链长度为  $K$  ( $K \leq n$ ) 的一段。现在有个苦逼的找不到工作的程序员想花最少的钱买到颜色最多的一段项链。要求时间复杂度  $O(N)$ ，空间复杂度  $O(1)$ 。

答案：看到这道题，第一时间想不到  $O(N)$  的思路不要慌，先来看看最裸的思路应该怎么写？枚举每一个买项链的位置，然后模拟一遍统计有多少个不同的珠子，代码是这样的：

[cpp] view plain copy

```
1. <span style="font-size:14px;">for (int i=0;i<n;i++) nek[i+n]=n  
   ek[i]; //首先把项链的长度扩大一倍，这样就把环拆成了一个链  
2. for (int i=0;i<n;i++) //枚举每一个开头  
3. {  
4.     int vis[10]; //vis 是用来统计每一个颜色的珠子出现的次数
```



```
5.  memset(vis,0,sizeof(vis));
6.  for (int j=i;j<i+k;j++)
7.  {
8.      vis[arr[j]]++;
9.  }
10. int num=0;
11. for (int j=1;j<=9;j++)
12.     if (vis[j]>0)
13.         num++;
14. ans=max(ans,num);
15. }
```

现在这样做，复杂度是  $O(N \cdot K)$  的，当  $K$  等于  $N$  的时候，复杂度就退化成了  $O(N \cdot N)$ 。所以要考虑如何降低一维复杂度，题目要求复杂度是  $O(N)$ ，因此，里面的  $K$  这个复杂度是有可能被降下来的。来考虑下面一个例子：

1 1 2 3 4

假设  $K=4$ ，那么  $[1,4]$  这个区间里面，vis 数组的值是：2 1 1 0，而  $[2,5]$  这个区间里面，vis 数组的值是  $[1,1,1,1]$ ，这个是怎么得到的呢？在  $[1,4]$  的 vis 的基础上， $\text{vis}[1]--$ ， $\text{vis}[4]++$ ，只进行了两次操作，就可以得到  $[2,5]$  区间的 vis 情况。所以，我们可以在一开始加入  $K$  个珠子进来，然后，每次，把最先进来的那个珠子  $\times$  踢出去，此时  $\text{vis}[x]--$ ，然后，再把下一个

获取更多资料礼包！

微信关注



珠子 Y 加进来，此时 `vis[y]++`，仅仅两次操作，就可以得到下一个位置的值了。代码写起来更方便：

[cpp] view plain copy

```
1. <span style="font-size:14px;">for (int i=0;i<n;i++) nek[i+n]=n
   ek[i]; //首先把项链的长度扩大一倍，这样就把环拆成了一个链
2. int vis[10]; //vis 是用来统计每一个颜色的珠子出现的次数
3. memset(vis,0,sizeof(vis));
4. int num=0;
5. for (int i=0;i<k-1;i++) //首先把前 k-1 个珠子加进来
6. {
7.     vis[arr[i]]++;
8.     if (vis[arr[i]]==1) //如果这个珠子是第一次出现，num++
9.         num++;
10.}
11.for (int i=k-1;i<n+k;i++) //枚举将每一个珠子 i 加进来
12.{
13.    //首先加入第 i 颗珠子
14.    vis[arr[i]]++;
15.    if (vis[arr[i]]==1)
16.        num++;
17.    ans=max(ans,num);
```



```
18. //然后把当前第一颗珠子删掉
19. vis[arr[i-(k-1)]]--;
20. if (vis[arr[i-(k-1)]]==0) //如果这颗珠子是自己颜色下唯一的那
    一颗
21. num--;
22. }
```

如此一来，时间复杂度从  $O(N \cdot K)$  降低到了  $O(N)$ ，空间复杂度是  $O(10) = O(1)$

三、输入一个整型数组，数组里有正数也有负数。

数组中连续的一个或多个整数组成一个子数组，每个子数组都有一个和。

求所有子数组的和的最大值。要求时间复杂度为  $O(n)$ 。

答案：典型的最大子段和问题，是动态规划 or 贪心的入门题目。在这里给一种贪心的做法：

从左到右不停的加，一旦当前的和小于 0，就把之前的所有元素都抛弃掉。

代码非常简单：

[cpp] view plain copy

```
1. <span style="font-size:14px;">int sum=0;
2. for (int i=0;i<n;i++)
3. {
```





```
4.  if (sum<0) sum=0;
5.  sum+=arr[i];
6.  ans=max(ans,sum);
7. }
```

四、假设：生男生女的概率都是 50%（实际情况是生男的概率稍大一些，暂不考虑）

如果：政策允许生下一个孩子，如果是男孩就不能再生，发现是女孩，可以继续生，直到生下来一个男孩不准再生了。

那么：以后的生男生女的比例会偏离 50%吗？偏向哪边？偏多少？说明理由

答案：做这种题目，千万不要被题面所迷惑，要坚信男女比例是不会因此而改变的！如果非要计算，也可以拿单个家庭作为研究对象来进行期望值的计算，可以发现其实这还是一个等比数列，最后生男生女的期望值都是相等的，都是 1 个，也就是说，在这种策略下平均每家要了两个孩子，一男一女。

计算方法如下：

生一个孩子就不生的概率是 0.5,生两个孩子就不生的概率是  $0.5 \times 0.5$ ，所以，

生孩子个数的期望值是：

$$1 \times 0.5 + 2 \times 0.5 \times 0.5 + 3 \times 0.5 \times 0.5 \times 0.5 + \dots = 2$$

生男孩的期望值是：

获取更多资料礼包！

微信关注：白熊事务所



$$1*0.5+1*0.5*0.5+1*0.5*0.5*0.5+\dots\dots=1$$

或者也可以写代码去做个实验：

[cpp] view plain copy

```
1. <span style="font-size:14px;">#include <iostream>
2. #include <time.h>
3. #include <stdlib.h>
4. using namespace std;
5. int n[2];
6. int main()
7. {
8.     srand(time(0));
9.     for (int i=1;i<=1000000000;i++)
10.    {
11.        int t=rand()%2;
12.        n[t]++;
13.        while (t!=0)
14.        {
15.            t=rand()%2;
16.            n[t]++;
17.        }
```



```
18.  }  
19.  cout<<n[0]<<' '<<n[1]<<endl;  
20.}  
21.</span>
```

五、袋子足够大，袋子里面有三种颜色的球。数量都是一样的，摸出来的概率是一样的。小明随便摸出来两个，然后看了一眼说至少有一个是红的。问，小明摸出来两个红球的概率是多少？？

答案：两个红色的组合数有一种，一红一其他的组合数有  $2+2=4$  种。

因此，两个红球的概率  $= 1 / (1+4) = 1/5$  种



icebear.me

白熊事务所致力为准备求职的小伙伴提供优质的资料礼包和高效的求职工具。礼包包括**互联网、金融等行业的求职攻略**；**PPT模板**；**PS技巧**；**考研资料**等。

微信扫码关注：**白熊事务所**，获取更多资料礼包。

登陆官网：**www.icebear.me**，教你如何**一键搞定名企网申**。

获取更多资料礼包！

微信关注：白熊事务所