

2. 战胜选择恐惧

热身晚会的第二个节目是抽奖。小明幸运的被抽中了，他被领到一排倒扣着的 100 个纸杯面前，纸杯盖住的是不同价值的奖品（巧克力？充值卡？校庆吉祥物？）。小明不知道纸杯下都是什么奖品，但当他看到奖品时，都会根据自己的喜好，给每个奖品打个分，不失一般性，设奖品的价值为 1, 2, ..., 100。纸杯下奖品的价值是均匀分布（Uniform Distribution）的。小明有两种不同的选择：

(1) 拿走任意一个纸杯下的奖品；(2) 从左往右依次打开纸杯，小明可以选择要或不要纸杯下的奖品。如果小明选择不要，可以继续打开后面的纸杯，但是他不能再选之前已经放弃的物品；如果小明选择要，则游戏结束。

小明是一个典型的选择困难者，你是他的好朋友，你能用 C 语言编写一个程序，模拟上述 2 个选择过程，帮他找到一个以最大概率获得最高价值的方案吗？

对于第(2)种选择，设奖品的总个数为 n ，一个思路是先打开前 k 个纸杯，并舍弃奖品（设前 k 个奖品的最高价值为 v ），然后在剩下的纸杯中选择一个价值大于 v 的奖品。很显然，如果最高价值在前 k 个奖品中，则一定不会选中最高价值的奖品，反之，选中最高价值的概率为多少呢？

我们可在奖品的 10,000 个均匀分布上模拟上述选择过程。对于每个分布，分别让 k 等于 1 到 99，看能否选中最高价值。对于每个 k ，记下整个过程中可选中最高价值的次数，然后看 k 取多少时，能够拿到最高价值的概率最大？这种模拟称为蒙特卡洛方法，是以位于摩纳哥的著名赌城 Monte Carlo 命名的 😊 (80 分)

这个问题实际上称为[苏格拉底麦穗问题](#)或者[秘书问题](#)。你能用概率相关理论证明计算机找到的方案是正确的吗？ (20 分)

```
1  #include <random>
2  #include <algorithm>
3  #include <iterator>
4  #include <iostream>
5  #include <vector>
6
7  int main() {
8      std::vector<int> v;
9      for (int i = 0; i < 100; i++){
10         v.push_back(i + 1);
11     }
12     std::random_device rd;
13     std::mt19937 mt(rd());
14     std::shuffle(v.begin(), v.end(), mt);
15     for (int i = 0; i < 100; i++){
16         std::cout << v[i] << " ";
17     }
18     std::cout << std::endl;
19     return 0;
20 }
```

图 1. 洗牌，随机打乱数组元素的位置