蓄水池采样算法

## 1.背景

* 1. 有一个大小为N的数组，N未知，需要从这N个数中采样出k个数
  2. 只能顺序从1开始读取每个数

## 2.算法

//构建一个大小为k的蓄水池

Init : a reservoir with the size： k

//从k+1开枚举到N

for i= k+1 to N

// 从i个样本里随机采样一个数M

M=random(1, i);

//如果M < k

if( M < k)

//替换蓄水池中第M个值为i

SWAP the Mth value and ith value

end for

具体描述如下：：先将前k个数取出来放入结果集中，然后从第k+1个数开始遍历。假设遍历到第i个数，以k/i的概率替换蓄水池中的样本即可

## 3.证明每个元素出现的概率都是相同的, 即k/n,

当 i < k 时，读取第i个数，每个数被选中的概率是1/i

当i > k时， 从 i = k+1开始， 前i-1个数被选中的概率是k/(i-1), 读第i个数时，

第一个数被选中的概率为：

上一次（i-1）被保留（没有被替换的概率） k/(i-1),

本次(i) 被保留的概率 1 – i/k

故当读取第i个数（i>k）时，每个数被保留的概率为 **k/(i-1) \* (1-k/i \* 1/k) = k/i,**

P1 = k/(i-1), 表示上一次被保留的概率， (1-k/i \* 1/k)表示本次不被替换的概率

## 4.代码实现

vector<int> sampling1(int N, int K) {

//设置随机数种子，使每次产生的随机序列不同

srand(time(NULL));

if (N < 1 || K < 1 || N < K)

{

return {};

}

//简化

vector<int> arr;

for (int i = 1; i <= N; i++) {

if (i <= K) {

arr.push\_back(i);

} else {

int j = rand()%i + 1;

//以1/i的生成一个随机数j，j < k (概率为k/i), 让[1...k]中的第j个指针指向第i个数

if (j <= K) {

arr[j] = i;

}

//否则跳过第i个数

else {

}

}

}

return arr;

}

**测试：从[1,100]中随机选择10个数**

//1 53 91 40 98 6 56 8 43 96

vector<int> res = sampling1(100,10);

参考：<https://blog.csdn.net/bitcarmanlee/article/details/52719202>

https://www.bilibili.com/video/BV17i4y1j7wE/?spm\_id\_from=333.337.search-card.all.click&vd\_source=843e98298bc70b0e35331918314486ce