Generating a Random Cyclic Permutation[0]

David Gries, Jinyun Xue[1]

Computer Science Department
Cornell University
Ithaca, NY 14853

TR86-786
September 1986

Introduction

We prove correct an algorithm that, given $n > 0$, stores in array $b(0..n-1)$ a random cyclic permutation of the integers in $0..n-1$, with each cyclic permutation having equal probability of being stored in b. The algorithm was developed by Sattolo [0]; our contribution is to present a proof that is somewhat more convincing.

Preliminaries

A permutation Π of a set S is a one-to-one function $\Pi: S \rightarrow S$. The values of S can be partitoned into *cycles*; for each value $j \in S$, the values $\{j, \Pi.j, \Pi^2.j, \Pi^3.j \ ...\}$ form a cycle. (We use the period "." for function application.) A permutation is cyclic if it consists of a single cycle.

There are several ways to represent a permutation Π of $0..n-1$ in an array b; here, we let $b.i = \Pi.i$ for each i in $0..n-1$. When dealing with sequences of integers in the paper, capital letters denote sequences of elements and small letters elements. Catenation is denoted by juxtaposition.

The algorithm and its proof

Let execution of the statement $random(r)$ assign to r a random number uniformly distributed and satisfying $0 \leq r < 1$ and function $floor.x$ yield the integer part of x, for $x \geq 0$.

We present Sattolo's algorithm and then argue about its correctness.

[0] This research was supported by the NSF under grant DCR-8320274.

[1] Visiting Cornell from the Computer Science Department, Jiangxi Normal University, Nanchang, People's Republic of China.

```
{n > 0}
for (i: 0 ≤ i < n: b.i := i) ;
i := n-1;
do i ≠ 0 → random(r);            {0 ≤ r < 1}
           s := floor.(i*r);     {0 ≤ s < i < n-1}
           b.i, b.s := b.s, b.i;
           i := i-1              {0 ≤ s ≤ i < n-1}
od
```

Execution of the algorithm, we claim, terminates with $b(0..n-1)$ a cyclic permutation of $0..n-1$ and with all cyclic permutations being equally likely. We begin our proof of this claim by presenting the first part $P0$ of our loop invariant:

$$P0:\ 0 \leq i < n \quad \wedge \quad perm(b, 0..n-1)$$

where $perm(b, c)$ means that sequence b is a permutation of set c. Left to the reader are the simple proofs that execution of the first two statements of the algorithm establishes *P0*, that each iteration maintains *P0*, and that the loop terminates (after exactly $n-1$ iterations). Hence, the algorithm terminates with b a permutation of $0..n-1$.

The second conjunct of the loop invariant, *P1*, will be used to show that upon termination the permutation in b is cyclic:

P1: b contains $i+1$ cycles $\wedge$
the values of $b(0..i)$ are in $i+1$ different cycles of the permutation

Initially, $i = n-1$ and b contains n singleton cycles, so *P1* is true.

We now show that *P1* is maintained by a loop iteration. By *P1*, the values $b.i$ and $b.s$ are in different cycles. Hence, by the following Lemma 0, which is proved at the end of the paper, swapping $b.i$ and $b.s$ merges their cycles into one cycle, thus reducing the number of cycles by 1. After the swap, the values of $b(0..i-1)$ are still in $i-1$ different cycles. Hence, reducing i by 1 reestablishes *P1*.

Upon termination, we have *P0*, *P1*, and $i = 0$; these together imply that b contains a single cycle and hence is cyclic.

(0) **Lemma**. Exchanging two elements from different cycles of a permutation merges those two cycles into one cycle. □

We now know that the algorithm terminates with b a cyclic permutation of $0..n-1$. We have to prove that each cyclic permutation has the same probability of being in b upon termination —assuming that *random.r* chooses a value between 0 and 1 with all values being equally likely.

Each iteration of the loop stores a value in s. By an *s-sequence* we mean the sequence of values s_0, ..., s_{n-1} stored in s during an execution of the algorithm, with s_i being stored in s during iteration i. Value s_0 is chosen from $0..n-2$, with all values being equally likely; s_1 is chosen from $0..n-1$, with all values being equally likely, and so forth, with s_{n-1} being chosen from 0..0, with all values (just one) being equally likely. Therefore, there are exactly $(n-1)!$ different s-sequences, with all being equally likely.

Coincidently, there are $(n-1)!$ cyclic permutations of $0..n-1$ —this fact comes directly from Cauchy's formula [2, pp122-123]. Therefore, our desired result follows if different s-sequences result in different permutations in $b(0..n-1)$, which is the following lemma.

(1) **Lemma 1.** Two executions of the algorithm that result in different s-sequences terminate with different permutations in $b(0..n-1)$.

Proof. For two different s-sequences, there exists a k such that the s-sequences have the same values $s_0, ..., s_{k-1}$ but have different values for s_k. Thus, for the two executions, after k iterations of the loop the values in $b(0..n-1)$ are the same. Because the two values s_k are different, however, the next iteration for the two executions places different values in $b.(n-(k+1))$. Since $b.(n-(k+1))$ is not changed by future iterations, the values in $b.(n-(k+1))$ remain different for the two executions, which means that the resulting permutations are different. □

Proof of Lemma 0.

(0) **Lemma**. Exchanging two elements from different cycles of a permutation merges those two cycles into one cycle.

Proof. A permutation Π —e.g. $\{(0,1),(1,2),(2,0)\}$— can be represented as $H\ /\ K$, where the two sequences H and K are its domain and range, with corresponding domain-range pairs appearing in corresponding positions of H and K (e.g. 0 1 2 / 1 2 0). We sometimes write this in a two-line form, as shown below. In this representation, columns can be interchanged without changing the permutation.

$$\begin{pmatrix} H \\ K \end{pmatrix} = \begin{pmatrix} 0\ 1\ 2 \\ 1\ 2\ 0 \end{pmatrix} = \begin{pmatrix} 0\ 2\ 1 \\ 1\ 0\ 2 \end{pmatrix}$$

Now, a permutation is cyclic iff it has a representation $H\ /\ K$ in which K is H but rotated one element to the left. For example, the cyclic permutation $\{(0,1),(1,2),(2,0)\}$ can be written as 0 1 2 / 1 2 0. It is this property that we use in proving Lemma (0), to which we now turn.

Let the two disjoint cycles of Π be written as follows, where p and q are arbitrary elements of the two cycles (note that in each the bottom row is the top but rotated one element to the left; note also that X (or Y) is empty if the cycle has one element):

$$\begin{pmatrix} p & X \\ X & p \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} q & Y \\ Y & q \end{pmatrix}$$

Since the cycles are disjoint, we can write them as the single permutation

$$\begin{pmatrix} p & X & q & Y \\ X & p & Y & q \end{pmatrix}$$

Exchanging the two elements p and q yields the permutation

$$\begin{pmatrix} p & X & q & Y \\ X & q & Y & p \end{pmatrix}$$

Since the bottom row is the top row rotated one element to the left, the permutation is a single cycle. □

References

[0] Sattolo, S. An algorithm to generate a random cyclic permutation. IPL 22 (May 1986), 315-317.

[1] Gries, D. *The Science of Programming.* Springer Verlag, New York, 1981.

[2] Berge, C. *Principles of Combinatorics.* Academic Press, New York and Landon, 1971.

[3] Feijen, W.H.J., A.J.M. van Gasteren, and D. Gries. In-situ inversion of cyclic permutation. TR85-703, Computer Science Department, Cornell University (accepted for publication in IPL).

[4] Xue, J., and Gries, D. Developing a linear algorithm for cubing a cyclic permutation. TR86-780, Computer Science Department, Cornell University.