



HAL
open science

Random Uniform Permutations

Lucas Gerin

► **To cite this version:**

| Lucas Gerin. Random Uniform Permutations. Master. Unknown Region. 2018. hal-03482893

HAL Id: hal-03482893

<https://cel.hal.science/hal-03482893v1>

Submitted on 16 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - ShareAlike 4.0 International License

MINI-COURSE: RANDOM UNIFORM PERMUTATIONS

Lucas Gerin, École Polytechnique (Palaiseau, France)
lucas.gerin@polytechnique.edu

This course is at the interplay between Probability and Combinatorics. It is intended for Master students with a background in Probability (random variables, expectation, conditional probability).

The question we will address is "What can we say about a *typical* large permutation?": the number of cycles, their lengths, the number of fixed points,... This is also a pretext to present some universal phenomena in Probability: reinforcement, the Poisson paradigm, size-bias,...

Contents

1	Brief reminder on permutations	1
2	How to simulate a random uniform permutation?	2
3	Typical properties of a random uniform permutation	6
4	How to sort S_n efficiently: average-case analysis of Quicksort	12

1 Brief reminder on permutations

Before we turn to *random* permutations, we will give a few definitions regarding non-random (or *deterministic* permutations).

A *permutation* of size $n \geq 1$ is a bijection $\sigma : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$. For example

$$\begin{array}{cccc} 1 & 2 & 3 & 4 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 2 & 4 & 3 & 1 \end{array}$$

is a permutation of size 4. In these notes we often write a permutation with its one-line representation $\sigma(1)\sigma(2)\dots\sigma(n)$. For example the above permutation is simply written 2431.

There are $n!$ permutations of size n .

Cycle decomposition

For our purpose, there is a convenient alternative way to encode a permutation: by its *cycle decomposition*. A *cycle* is a finite sequence of distinct integers, defined up to the cycle order. This means that the three following denote the same cycle:

$$(8, 3, 4) = (3, 4, 8) = (4, 8, 3),$$

while $(8, 3, 4) \neq (8, 4, 3)$.

The *cycle decomposition* of a permutation σ is defined as follows. We give the theoretical algorithm and detail the example of

1	2	3	4	5	6	7
↓	↓	↓	↓	↓	↓	↓
6	3	1	5	7	2	4

Algorithm

Start with 1st cycle (1)
 Add to this cycle $\sigma(1)$, then $\sigma(\sigma(1))$, then $\sigma(\sigma(\sigma(1)))$,
 and so on until one of this number is one.
 Start the 2d cycle with a number which has not been
 seen before.
 Complete the 2d cycle with same procedure.
 Create new cycles until there is no remaining number.

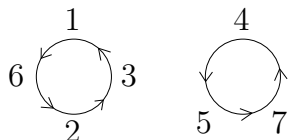
Finally, the cycle decomposition of σ is

$$(1, 6, 2, 3), (4, 5, 7)$$

Example

(1)
 $(1) \rightarrow (1, 6) \rightarrow (1, 6, 2) \rightarrow (1, 6, 2, 3)$
 and the cycle is over since $\sigma(3) = 1$.
 1st cycle (1, 6, 2, 3). 2d cycle: (4)
 1st cycle (1, 6, 2, 3). 2d cycle: $(4) \rightarrow (4, 5) \rightarrow (4, 5, 7)$.
 Done.

It is convenient to represent the cycle decomposition of σ with the following diagram:



Exercise 1 What is the cycle decomposition of 62784315?

2 How to simulate a random uniform permutation?

We will first discuss the following question. Imagine that you are given a random number generator `rand` (in your favourite programming language) which returns independent uniform random variables. How to use `rand` to simulate a random uniform permutation of size n ?

2.1 The naive algorithm

It works as follows:

- Pick $\sigma(1)$ uniformly at random in $\{1, 2, \dots, n\}$ (n choices);
- Pick $\sigma(2)$ uniformly at random in $\{1, 2, \dots, n\} \setminus \{\sigma(1)\}$ ($n - 1$ choices);
- Pick $\sigma(3)$ uniformly at random in $\{1, 2, \dots, n\} \setminus \{\sigma(1), \sigma(2)\}$ ($n - 2$ choices),

and so on until $\sigma(n)$ (1 choice).

By construction every permutation occurs with probability $1/n!$ so the output is uniform.

2.2 The "continuous" algorithm

- Pick continuous random variables X_1, X_2, \dots, X_n , independently and uniformly in $(0, 1)$;
- With probability 1 the n values are pairwise distinct. Therefore there exists a unique permutation σ such that

$$X_{\sigma(1)} < X_{\sigma(2)} < X_{\sigma(3)} < \dots < X_{\sigma(n)}.$$

- This σ is your output.

Proposition 1. *For every n , the output of the continuous algorithm is uniform among the $n!$ permutations of size n .*

Proof. Step 1: The n values are distinct. We have to prove that

$$\mathbb{P}(\text{for all } i \neq j, X_i \neq X_j) = 1.$$

We prove that the complement event $\{\text{there are } i, j \text{ such that } X_i = X_j\}$ has probability zero. First we notice that

$$\begin{aligned} \mathbb{P}(\text{there are } i \neq j \text{ such that } X_i = X_j) &= \mathbb{P}(\cup_{i \neq j} \{X_i = X_j\}) \\ &\leq \sum_{i \neq j} \mathbb{P}(X_i = X_j), \end{aligned}$$

by the union bound⁽ⁱ⁾. Now,

$$\mathbb{P}(X_i = X_j) = \int_{(0,1)^2} \mathbf{1}_{x=y} dx dy = \int_{y \in (0,1)} \left(\int_{x \in (0,1)} \mathbf{1}_{x=y} dx \right) dy = \int_{y \in (0,1)} \left(\int_{x=y}^y dx \right) dy = \int_{y \in (0,1)} 0 \times dy = 0.$$

Step 2: The output σ is uniform. To avoid messy notations we make the proof in the case $n = 3$. Since the 3 values X_1, X_2, X_3 are distinct we have

$$\begin{aligned} 1 &= \mathbb{P}(X_1 < X_2 < X_3) + \mathbb{P}(X_1 < X_3 < X_2) + \mathbb{P}(X_2 < X_1 < X_3) \\ &\quad + \mathbb{P}(X_2 < X_3 < X_1) + \mathbb{P}(X_3 < X_1 < X_2) + \mathbb{P}(X_3 < X_2 < X_1) \\ &= \int_{(0,1)^3} \mathbf{1}_{x_1 < x_2 < x_3} dx_1 dx_2 dx_3 + \int_{(0,1)^3} \mathbf{1}_{x_1 < x_3 < x_2} dx_1 dx_2 dx_3 + \int_{(0,1)^3} \mathbf{1}_{x_2 < x_1 < x_3} dx_1 dx_2 dx_3 \\ &\quad + \int_{(0,1)^3} \mathbf{1}_{x_2 < x_3 < x_1} dx_1 dx_2 dx_3 + \int_{(0,1)^3} \mathbf{1}_{x_3 < x_1 < x_2} dx_1 dx_2 dx_3 + \int_{(0,1)^3} \mathbf{1}_{x_3 < x_2 < x_1} dx_1 dx_2 dx_3. \end{aligned}$$

Now, x_1, x_2, x_3 are dummy variables in the above integrals, so they are interchangeable. Therefore, these 6 integrals are identical and each of these is $1/6 = 1/3!$. \square

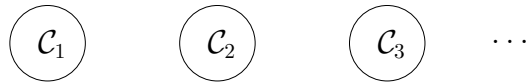
2.3 The "Chinese restaurant" algorithm

We introduce the Chinese restaurant algorithm, also called the Fisher-Yates algorithm (or even Fisher-Yates-Knuth algorithm). The main difference with the two previous algorithms is that the output σ will be described through its cycle decomposition.

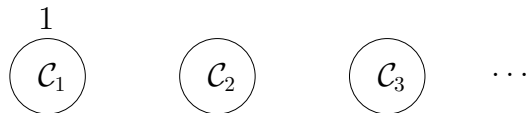
The algorithm runs as follows:

⁽ⁱ⁾The union bound says that $\mathbb{P}\left(\bigcup_{n \geq 1} A_n\right) \leq \sum_{n \geq 1} \mathbb{P}(A_n)$ for every sequence of events (A_n) .

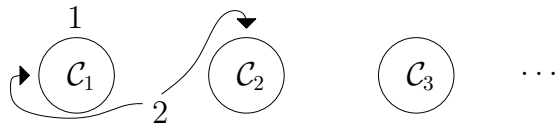
- Assume we are given infinitely many "restaurant tables" $\mathcal{C}_1, \mathcal{C}_2, \dots$. These tables are large enough so that an arbitrary number of people can sit at each table.



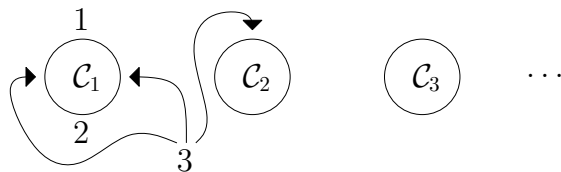
- Infinitely many customers $1, 2, 3, \dots$ enter the restaurant, one at a time. Put Customer $n.1$ at table \mathcal{C}_1 :



- With equal probability one-half, put Customer $n.2$ either at the same table as 1 (on its right) or alone at the new table \mathcal{C}_2 :

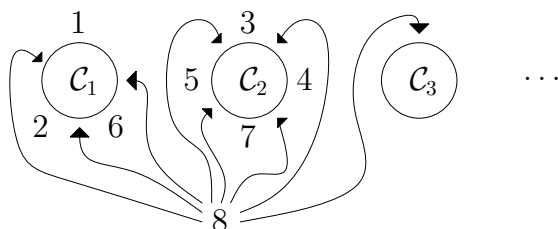


- With equal probability one-third, put Customer $n.3$ either on the right of 1, or on the right of 2, or alone at the first empty table:

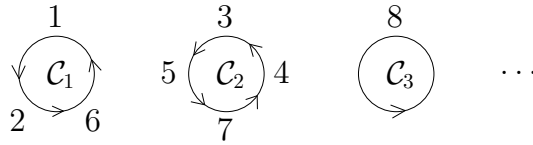


- ...

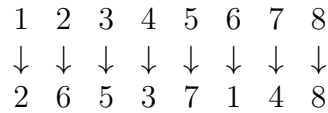
- Assume that customers $1, 2, \dots, n-1$ are already installed. With equal probability $1/n$, put Customer n either on the right of $1, \dots$, or on the right of $n-1$, or alone at the first empty table (here $n=8$):



Now, we return the permutation σ whose cycle decomposition corresponds to table repartitions. Assume here that 8 sits alone, we obtain the diagram



This can also be written $(126)(3547)(8)$. The corresponding permutation is



Exercise 2 Take $n = 4$. What is the probability that the output of the algorithm is the permutation 4231? (Hint: First write the cycle decomposition of 4231.)

Proposition 2. For every n , the output of the Chinese restaurant algorithm is uniform among the $n!$ permutations of size n .

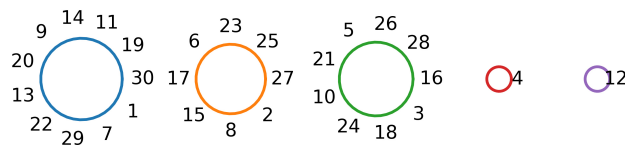
Proof. By construction, each table repartition with n customers occurs with the same probability

$$1 \times \frac{1}{2} \times \frac{1}{3} \times \cdots \times \frac{1}{n}.$$

Now, each table repartition corresponds to exactly one permutation of size n . Therefore each permutation occurs with probability $1/n!$. \square

Simulations

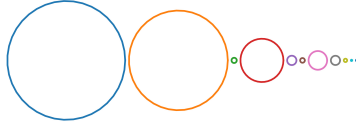
Here is a simulation for $n = 30$:



Here is a simulation for $n = 2000$ (We only represent sizes of tables. They have respective sizes 122, 673, 631, 68, 176, 159, 35, 8, 28, 91, 2, 5, 1, 1.):



A last simulation for $n = 30000$. Tables have sizes 15974, 11238, 31, 2121, 99, 25, 397, 97, 13, 2, 3.



For more on the Chinese restaurant we refer to [5]. On the following webpage you can run simulations of the Chinese restaurant by yourself:

<http://gerin.perso.math.cnrs.fr/ChineseRestaurant.html>

3 Typical properties of a random uniform permutation

From now on S_n denotes a random uniform permutation of size n , generated by any of the previous algorithms.

3.1 Number of fixed points

Definition 1. Let σ be a permutation of size n . The integer $1 \leq i \leq n$ is a fixed point of σ if $\sigma(i) = i$.

For example, 2431 has a unique fixed point at $i = 3$.

Proposition 3. Let F_n be the number of fixed points of S_n . For every n , we have that⁽ⁱⁱ⁾

$$\mathbb{E}[F_n] = 1, \quad \text{Var}(F_n) = 1.$$

This is quite surprising that $\mathbb{E}[F_n]$ and $\text{Var}(F_n)$ do not depend on n .

Proof. We write $F_n = \sum_{i=1}^n X_i$, where

$$X_i = \begin{cases} 1 & \text{if } S_n(i) = i, \\ 0 & \text{otherwise} \end{cases}.$$

Random variables X_i 's are *not* independent. Still we have by linearity of expectation that

$$\mathbb{E}[F_n] = \mathbb{E}[X_1 + \cdots + X_n] = \mathbb{E}[X_1] + \cdots + \mathbb{E}[X_n],$$

and we are left to compute $\mathbb{E}[X_i]$ for every i . Now,

$$\mathbb{P}(X_i = 1) = \mathbb{P}(S_n(i) = i) = \frac{\text{card}\{\text{permutations } s \text{ of size } n \text{ with } s(i) = i\}}{\text{card}\{\text{permutations of size } n\}} = \frac{(n-1)!}{n!} = \frac{1}{n}.$$

(Indeed, a permutation such that $s(i) = i$ is also a permutation of the set $\{1, 2, \dots, i-1, i+1, \dots, n\}$ of size $n-1$.) Therefore we have that

$$\mathbb{E}[X_i] = 1 \times \mathbb{P}(X_i = 1) + 0 \times \mathbb{P}(X_i = 0) = 1/n.$$

⁽ⁱⁱ⁾Thank you to Amic Frouvelle for pointing me that the variance was wrong in the previous version of these notes.

Finally

$$\mathbb{E}[F_n] = \mathbb{E}[X_1] + \cdots + \mathbb{E}[X_n] = n \times 1/n = 1.$$

In order to compute the variance we will use the formula

$$\begin{aligned} \text{Var}\left(\sum X_i\right) &= \sum_i \text{Var}(X_i) + \sum_{i \neq j} \text{Cov}(X_i, X_j) \\ &= n\text{Var}(X_1) + n(n-1)\text{Cov}(X_1, X_2) \end{aligned}$$

By the previous computation we have:

$$\mathbb{E}[X_1] = \frac{1}{n}, \quad \text{Var}(X_1) = \frac{1}{n}\left(1 - \frac{1}{n}\right).$$

Similarly as above we can compute

$$\mathbb{E}[X_1 X_2] = \mathbb{P}(X_1 \times X_2 = 1) = \mathbb{P}(X_1 = 1, X_2 = 1) = \frac{(n-2)!}{n!} = \frac{1}{n(n-1)}.$$

Hence $\text{Cov}(X_1, X_2) = \frac{1}{n(n-1)} - \mathbb{E}[X_1]\mathbb{E}[X_2] = \frac{1}{n(n-1)} - \frac{1}{n^2}$. Finally

$$\text{Var}(F_n) = 1 - \frac{1}{n} + n(n-1)\frac{1}{n^2(n-1)} = 1.$$

□

The Poisson paradigm

There is a general phenomenon in probability known as the *Poisson paradigm*. It says that if X_i 's are 0/1 random variable such that

1. $\mathbb{E}[X_i] = \mathbb{P}(X_i = 1)$ is "small" for every i ;
2. X_i 's are "almost" independent ;

then $X = \sum X_i$ is almost distributed like the Poisson distribution with mean $\sum \mathbb{E}[X_i]$. Here $\sum \mathbb{E}[X_i] = \sum_{i=1}^n 1/n = 1$ and one can make the Poisson paradigm rigorous:

Proposition 4. *Let $(S_n)_n$ be a sequence of random uniform permutations, and let F_n be the number of fixed points of S_n . Then F_n converges in distribution to the Poisson distribution with mean 1, i.e.*

$$\mathbb{P}(F_n = k) \xrightarrow{n \rightarrow +\infty} \mathbb{P}(\text{Poisson}(1) = k) = \frac{e^{-1}}{k!},$$

for every $k = 0, 1, 2, \dots$

A combinatorial proof can be found at [8]. For more on the Poisson paradigm, we refer to [2].

3.2 Number of inversions

An *inversion* in σ is a pair (i, j) such that

$$\begin{cases} i < j, \\ \sigma(i) > \sigma(j). \end{cases}$$

Let $\text{Inv}(\sigma)$ be the number of inversions of σ . For example, if $\sigma = 43152$ then $\text{Inv}(\sigma) = 6$ (each arc counts for an inversion):

$$\sigma: \quad \begin{array}{cccccc} & \frown & \frown & \frown & \frown & \\ & 4 & 3 & 1 & 5 & 2 \end{array}$$

Proposition 5. For every n , let S_n be a uniform random permutation of size n . Then

$$\mathbb{E}[\text{Inv}_n(S_n)] = \frac{n(n-1)}{4}.$$

Proof. We will make a combinatorial proof, with (almost) no computation. First, let $\tilde{\sigma}$ be the *reversed* permutation of σ : for every $1 \leq i \leq n$,

$$\tilde{\sigma}(i) = n + 1 - \sigma(i).$$

For instance, if $\sigma = 43152$ then $\tilde{\sigma} = 23514$. Then by construction we have that an arbitrary pair (i, j) is an inversion for σ if and only if it is not an inversion for $\tilde{\sigma}$. We deduce that

$$\text{Inv}(\sigma) + \text{Inv}(\tilde{\sigma}) = \text{card} \{ \text{all pairs } 1 \leq i < j \leq n \} = \binom{n}{2} = \frac{n(n-1)}{2}.$$

Here we see that $\text{Inv}(43152) + \text{Inv}(23514) = 6 + 4 = \binom{5}{2}$:

$$\begin{array}{c} \sigma: \quad \begin{array}{cccccc} & \frown & \frown & \frown & \frown & \\ & 4 & 3 & 1 & 5 & 2 \end{array} \\ \\ \tilde{\sigma}: \quad \begin{array}{cccccc} & \frown & \frown & \frown & \frown & \\ & 2 & 3 & 5 & 1 & 4 \end{array} \end{array}$$

Now, we apply the above equality to $\sigma = S_n$ and take expectations of both sides:

$$\mathbb{E}[\text{Inv}(S_n)] + \mathbb{E}[\text{Inv}(\tilde{S}_n)] = \frac{n(n-1)}{2}.$$

But now, it is obvious that $\sigma \mapsto \tilde{\sigma}$ is a bijection so it preserves the uniform measure. Therefore \tilde{S}_n is also a uniform random permutation and we have $\mathbb{E}[\text{Inv}(S_n)] = \mathbb{E}[\text{Inv}(\tilde{S}_n)]$. The proof is done. \square

3.3 Number of cycles

Proposition 6. Let C_n be the number of cycles of S_n . When $n \rightarrow +\infty$,

$$\mathbb{E}[C_n] \stackrel{n \rightarrow +\infty}{\sim} \log(n).$$

Proof. We may assume that S_n is the output of the Chinese restaurant algorithm. All along the process of the Chinese restaurant, a new cycle appears when a customer sits at a new table:

$$C_n = \sum_{i=1}^n Z_i,$$

where

$$Z_i = \begin{cases} 1 & \text{if Customer } i \text{ sits at a new table,} \\ 0 & \text{otherwise} \end{cases}.$$

Customer i sits at a new table with probability $1/i$, therefore $\mathbb{E}[Z_i] = 1/i$. Then,

$$\mathbb{E}[C_n] = \mathbb{E} \left[\sum_{i=1}^n Z_i \right] = \sum_{i=1}^n \mathbb{E}[Z_i] = \sum_{i=1}^n \frac{1}{i}.$$

Now, we use the fact that⁽ⁱⁱⁱ⁾ $\sum_{i=1}^n \frac{1}{i} \sim \log(n)$. □

3.4 Size of the first cycle/first table

Let $T_1(n)$ be the number of customers at Table 1 in the Chinese restaurant process at time n . By Proposition 2, we have that the random variable $T_1(n)$ has the distribution of the cycle of 1 in the cycle decomposition of a random uniform permutation of size n .

Proposition 7. For every n , the random variable $T_1(n)$ is uniformly distributed in $\{1, 2, \dots, n\}$, i.e.

$$\mathbb{P}(T_1(n) = i) = \frac{1}{n}, \quad \text{for every } i \in \{1, 2, \dots, n\}.$$

Remark . 1. The distribution of the sequence $(T_1(n))_{n \geq 1}$ is actually known as the Pólya Urn process [6].

2. A nice problem related to Proposition 7 is given by the 100 prisoners problem [7].

Proof. 1st proof: Probability.

The proof goes by induction. For $n = 1$ this is obvious since with probability one $T_1(1) = 1$.

Assume now that for some $n \geq 1$, the random variable $T_1(n)$ is uniform in $\{1, 2, \dots, n\}$. If $T_1(n) = i$, then Customer $n + 1$ sits at table 1 with probability $i/(n + 1)$.

⁽ⁱⁱⁱ⁾See [https://en.wikipedia.org/wiki/Harmonic_series_\(mathematics\)](https://en.wikipedia.org/wiki/Harmonic_series_(mathematics))

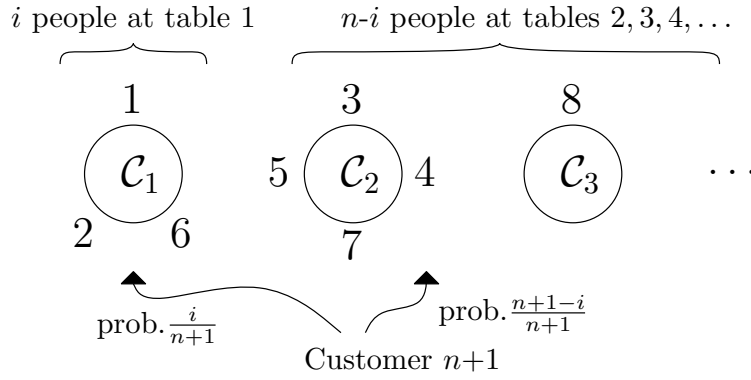


Figure: A sketch of the situation when Customer $n + 1$ tries to sit.

Therefore

$$T_1(n+1) = \begin{cases} i+1 & \text{with probab. } \frac{i}{n+1}, \\ i & \text{with probab. } \frac{n+1-i}{n+1}. \end{cases} \quad (1)$$

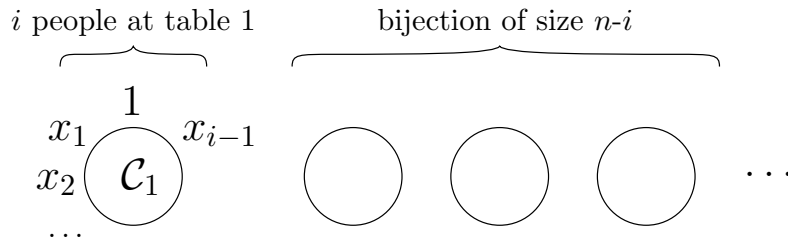
Fix $j \in \{1, \dots, n+1\}$. The above argument implies that

$$\begin{aligned} \mathbb{P}(T_1(n+1) = j) &= \mathbb{P}(T_1(n+1) = j \cap T_1(n) = j) + \mathbb{P}(T_1(n+1) = j \cap T_1(n) = j-1) \\ &= \mathbb{P}(T_1(n+1) = j | T_1(n) = j) \mathbb{P}(T_1(n) = j) \\ &\quad + \mathbb{P}(T_1(n+1) = j | T_1(n) = j-1) \mathbb{P}(T_1(n) = j-1) \\ &= \frac{n+1-j}{n+1} \times \mathbb{P}(T_1(n) = j) \quad (\text{apply (1) with } i = j.) \\ &\quad + \frac{j-1}{n+1} \times \mathbb{P}(T_1(n) = j-1) \quad (\text{apply (1) with } i = j-1.) \\ &= \frac{n+1-j}{n+1} \times \frac{1}{n} + \frac{j-1}{n+1} \times \frac{1}{n} \quad (\text{recall } T_1(n) \text{ is uniform}) \\ &= \frac{n}{(n+1)n} = \frac{1}{n+1}, \end{aligned}$$

which proves that $T_1(n+1)$ is uniform in $\{1, \dots, n+1\}$.

2d proof: Combinatorics.

For $i = 1, \dots, n$, let us enumerate the permutations in which $T_1(n) = i$. We have to choose $i-1$ elements x_1, \dots, x_{i-1} ($\binom{n-1}{i-1}$ choices) which belong to this cycle, and put them in a given order ($(i-1)!$ choices). Then, the $n-i$ remaining elements form a permutation of size $n-i$ ($(n-i)!$ choices).



Therefore

$$\begin{aligned}\mathbb{P}(T_1(n) = i) &= \frac{\text{card}\{\text{permutations of size } n \text{ with } T_1(n) = i\}}{n!} \\ &= \frac{1}{n!} \binom{n-1}{i-1} (i-1)!(n-i)! \\ &= \frac{1}{n!} \frac{(n-1)!}{(i-1)!(n-i)!} (i-1)!(n-i)! = \frac{1}{n}.\end{aligned}$$

□

Discussion: the reinforcement phenomenon

The Chinese restaurant process illustrates the *reinforcement phenomenon* which is very common in Probability. It is also known as the "rich gets richer" phenomenon. Indeed, we observe that the more people there are at Table 1 at a given time, the more there will be in the future.

As an application, it turns out that because Table 1 appears sooner than Table 2, Table 1 is much more occupied (in average) than Table 2.

Proposition 8. *For large n , we have that*

$$\mathbb{E}[T_1(n)] \stackrel{n \rightarrow +\infty}{\sim} \frac{n}{2}, \quad \mathbb{E}[T_2(n)] \stackrel{n \rightarrow +\infty}{\sim} \frac{n}{4}.$$

Proof. First, we claim that conditionally on the event $\{T_1(n) = i\}$, then $T_2(n)$ is uniformly distributed in $\{1, 2, \dots, n - i\}$: for every $j \leq n - i$ we have

$$\mathbb{P}(T_2(n) = j \mid T_1(n) = i) = \begin{cases} \frac{1}{n-i} & \text{if } i < n, \\ 0 & \text{if } i = n. \end{cases}$$

We skip the proof, which is very similar to the proof of Proposition 7 (in this case the combinatorial proof is easier).

Consequently, if we condition on the event $\{T_1(n) = i\}$ we have that

$$\begin{aligned}\mathbb{E}[T_2(n) \mid T_1(n)] &= \mathbb{E}[\text{Uniform random var. in } \{1, 2, \dots, n - T_1(n)\}] \\ &= \frac{1 + n - T_1(n)}{2}.\end{aligned}$$

Now, by the tower property of conditional expectation^(iv) we obtain

$$\mathbb{E}[T_2(n)] = \mathbb{E}\left[\mathbb{E}[T_2(n) \mid T_1(n)]\right] = \mathbb{E}\left[\frac{1 + n - T_1(n)}{2}\right] = \frac{1 + n - n/2}{2} \sim \frac{n}{4}.$$

□

^(iv)This says that $\mathbb{E}[\mathbb{E}[X \mid Y]] = \mathbb{E}[X]$.

Discussion: the size-bias phenomenon

We conclude by investigating an apparent paradox:

- In average, there are $n/2$ people at the same table as 1. But recall that the output of the Chinese restaurant process is uniform in \mathfrak{S}_n so by symmetry, every element in $\{1, 2, \dots, n\}$ plays the same role: this table can be considered as a *typical* table.
- There are in average $\log(n)$ distinct tables, so a *typical* table should have (in average) about

$$\frac{\text{Number of customers}}{\text{Number of tables}} \approx \frac{n}{\log(n)} \ll \frac{n}{2}$$

customers.

The paradox is that Table 1 is *not* typical: by saying that 1 sits at this table the size of this table is biased. The size of Table 1 is overestimated compared to a "true" typical table. This is the *size-bias* phenomenon, whose a very nice introduction can be found in [1].

4 How to sort S_n efficiently: average-case analysis of Quicksort

We will discuss a different topic regarding random permutations: the analysis of sorting algorithms. We will focus on one of the most famous: **Quicksort**.

4.1 The algorithm

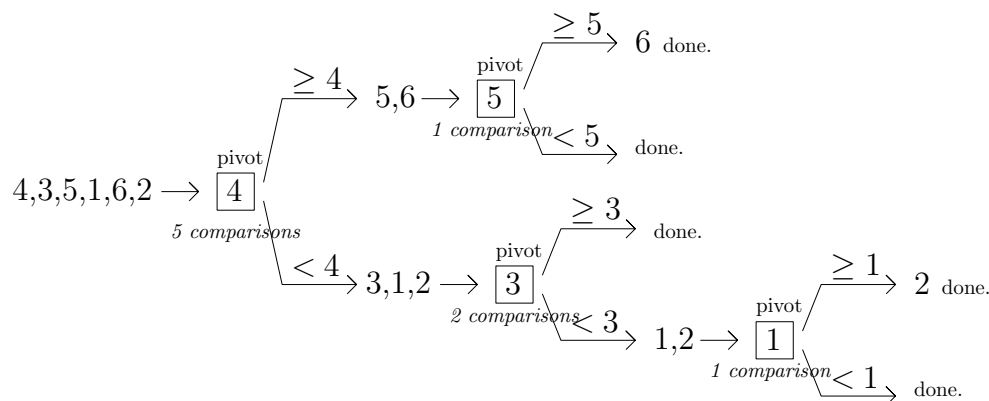
Input: Sequence of numbers x_1, x_2, \dots, x_n

Output: Re-ordered sequence $x_{\sigma(1)} \leq x_{\sigma(2)} \cdots \leq x_{\sigma(n)}$

The algorithm uses the *Divide-and-Conquer* strategy, there are three steps:

1. Call x_1 the *pivot* of the list.
2. Compare all the elements x_2, \dots, x_n with x_1 and re-order the list so that
 - (a) elements $< x_1$ come before the pivot,
 - (b) elements $\geq x_1$ come after the pivot.
3. Recursively apply strategy to both sub-lists.

Here are the first steps applied to the permutation 435162:



4.2 Average-case analysis

We consider that the cost of the algorithm driven on x_1, \dots, x_n is given by the number $\text{Comp}(x_1, \dots, x_n)$ of pairwise comparisons between x_i 's. For instance, in the above example we have that

$$\text{Comp}(4, 3, 5, 1, 6, 2) = 5 + 1 + 2 + 1 = 9.$$

If the input is random, then Comp is a random variable.

Proposition 9. *Let X_1, \dots, X_n be independent random variables uniform in the interval $(0, 1)$. Then, when $n \rightarrow +\infty$,*

$$\mathbb{E}[\text{Comp}(X_1, \dots, X_n)] = 2n \log(n) + o(n \log(n)).$$

Both the algorithm and its analysis were provided by Hoare [4]. A modern reference is [3].

Proof. By construction X_1 is the first pivot. Denote by Y_1, \dots, Y_{I-1} be the numbers $> X_1$, and Z_1, \dots, Z_{n-I} , so that I is the (random) rank of X_1 in the sequence. Because of the recursive strategy the number of comparisons is given by

$$\text{Comp}(X_1, \dots, X_n) = \underbrace{n-1}_{\text{Comp. with } X_1} + \text{Comp}(Y_1, \dots, Y_{I-1}) + \text{Comp}(Z_1, \dots, Z_{n-I}). \quad (\star)$$

We omit the proofs of the two following claims:

- The rank I is uniform in $1, 2, \dots, n$.
- Conditionally on X_1 , the Y_j 's are i.i.d. (and uniform in $(0, X_1)$) and the Z_j 's are i.i.d. (and uniform in $(X_1, 1)$).

Therefore, if we take expectations of both sides of (\star) and put $c_n = \mathbb{E}[\text{Comp}(X_1, \dots, X_n)]$ then we obtain

$$\begin{aligned} c_n &= n - 1 + \sum_{i=1}^n \mathbb{P}(I = i) (c_{i-1} + c_{n-i}) \\ &= n - 1 + \frac{1}{n} \sum_{i=1}^n c_{i-1} + \frac{1}{n} \sum_{i=1}^n c_{n-i} \\ &= n - 1 + \frac{2}{n} \sum_{i=1}^n c_{i-1}, \end{aligned}$$

with $c_0 = c_1 = 0$. In order to get rid of the sums we compute

$$\begin{aligned} nc_n - (n-1)c_{n-1} &= n(n-1) + 2 \sum_{i=1}^n c_{i-1} - (n-1)(n-2) - 2 \sum_{i=1}^{n-1} c_{i-1} \\ &= 2(n-1) + 2 \sum_{i=1}^n c_{i-1} - 2 \sum_{i=1}^{n-1} c_{i-1} \\ &= 2(n-1) + 2c_{n-1} \end{aligned}$$

so finally

$$nc_n = 2(n-1) + (n+1)c_{n-1}.$$

This can be rewritten as:

$$n(c_n + 2n) = 2n + (n+1)(c_{n-1} + 2(n-1)).$$

If we divide by $n(n+1)$ we get

$$\frac{c_n + 2n}{n+1} = \frac{2}{(n+1)} + \frac{c_{n-1} + 2(n-1)}{n}.$$

If we put $d_n := \frac{c_n + 2n}{n+1}$ we have that

$$d_n = \frac{2}{n+1} + \frac{2}{n} + \frac{2}{n-1} + \cdots + \frac{2}{5} + \frac{2}{4} + d_2.$$

i.e. $d_n = 2 \log(n) + o(\log(n))$. Finally,

$$c_n = 2n \log(n) + o(n \log(n)).$$

□

(We observe that the number of comparisons $\text{Comp}(X_1, \dots, X_n)$ only depends on the relative order of the X_i 's, not on their exact values. Therefore Proposition 9 remains true (with the same proof) if X_i 's are *i.i.d.* with an arbitrary density.)

References

- [1] R.Arratia, L.Goldstein. Size bias, sampling, the waiting time paradox, and infinite divisibility: when is the increment independent? Available at <https://arxiv.org/abs/1007.3910> (2010).
- [2] A.D.Barbour, L.Holst, S.Janson. *Poisson approximation*. Oxford Univ. Press (1992).
- [3] P.Flajolet, R.Sedgewick. *An introduction to the analysis of algorithms*. Addison-Wesley (1996).
- [4] C.A.Hoare. Quicksort. *The Computer Journal*, vol.5, n.1, p.10-16 (1962).
- [5] J.Pitman. *Combinatorial stochastic processes*. Lecture notes for the Saint-Flour summer school (available online) (2002).
- [6] N.Pouyanne. Pólya urn models. Proceedings of *Nablus'14 CIMPA Summer School: Analysis of Random Structures*, p.65-87. Available at <https://hal.archives-ouvertes.fr/hal-01214113/> (2014).
- [7] Wikipedia page of the *100 prisoners problem*. https://en.wikipedia.org/wiki/100_prisoners_problem.
- [8] Wikipedia page of *Rencontres numbers*. https://en.wikipedia.org/wiki/Rencontres_numbers.