

# Bucket Sort When You Know The Distribution.

David Ponarovsky

February 23, 2023

## Abstract

None.

**The problem.** Let  $f : [0, 1] \rightarrow [0, 1]$  a fixed distribution function. Write an algorithm that sorts  $n$  draws  $x_1 \dots x_n$  at linear expectation time.

**Solution.** We will define a partition of the input into a series of  $n$  buckets  $\mathcal{B} = \{B_k = [t_k, t_{k+1}] : k \in [n]\}$  such that  $\Pr[x \in B_i] = \frac{1}{n}$  for any bucket. Assume that we succeed in computing the buckets efficiently. Let the  $X_{ij}$  be the indicator of the event that  $x_j$  falls to  $B_i$ . Then we have:

$$\begin{aligned} \Pr \left[ \sum_i |B_i|^2 \geq t \right] &= \Pr \left[ \sum_i \left( \sum_j X_{ij} \right)^2 \geq t \right] \\ &= \Pr \left[ \sum_{i,j,j'} X_{i,j} X_{i,j'} \geq t \right] = \Pr \left[ \sum_{i,j \neq j'} X_{i,j} X_{i,j'} \geq t - n \right] \\ &\leq \frac{\sum_{i,j \neq j'} \mathbf{E}[X_{ij} X_{ij'}]}{t - n} = \frac{n}{(t - n)n^2} 2 \binom{n}{2} \leq \frac{n}{t - n} \end{aligned}$$

It follows that for any function  $t : \mathbb{N} \rightarrow \mathbb{R}$ , such that  $n = o(t)$ , sorting quadratic each bucket at turn would last almost surely less than  $t(n)$ . It shows that knowing the distribution enables one to compute the buckets efficiently. Ensuring the uniform partitioned property leads to the following recursive relation:

$$\begin{aligned} \frac{1}{n} &= \Pr[x \in B_k] = f(t_{k+1}) - f(t_k) \\ &\Rightarrow t_{k+1} \leftarrow f^{-1} \left( \frac{1}{n} + f(t_k) \right) \end{aligned}$$

Hence, if  $f$  can be computed in sublinear time, then we obtained an expected linear time algorithm for sorting  $\square$  The result above demonstrates a case when knowing how the input is distributed turns the problem equivalent to facing a uniform distributed input.