

A PROBABILISTIC COUNTING ALGORITHM

Marianne Durand & Philippe Flajolet

Projet ALGO–INRIA Rocquencourt

Séminaire Algo

THE PROBLEM

How to estimate the number of **distinct** elements in a large collection of data?

Given:

- One single pass
- Small memory
- Few calculations
- No assumptions on the distribution

Applications

- Data mining optimizations
- Routers

SURVEY OF THE ALGORITHMS

1. Hashing Schemes

[Estan-Varghese](#)

2. Adaptive Sampling

[Flajolet](#)

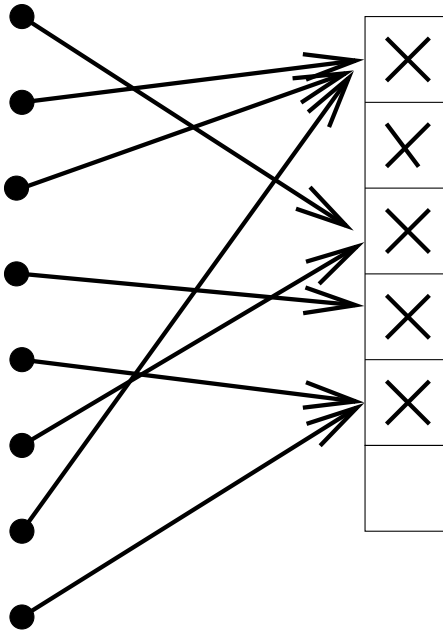
3. Probabilistic Counting

[Flajolet-Martin, Alon-Matias-Szegedy](#)

4. **New!** Maximum Based Probabilistic Counting

[Durand-Flajolet](#)

HASHING-HIT COUNTING



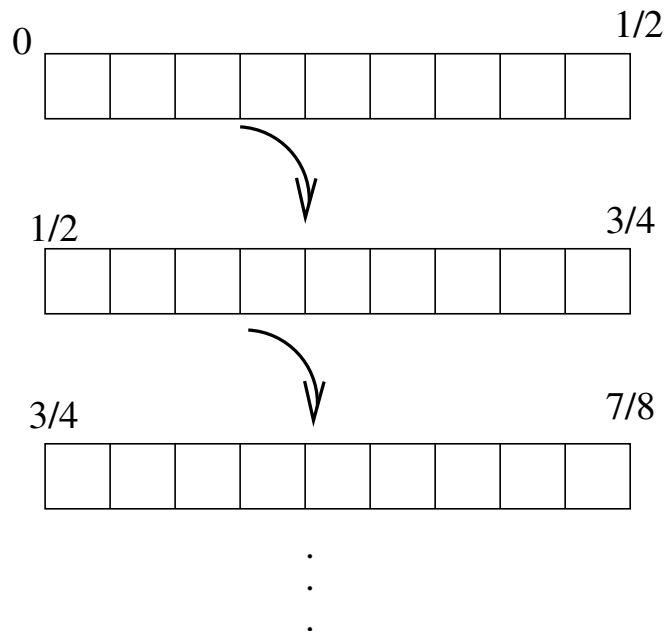
Question: given the hash table, estimate the number of distinct elements.

Answer: $-size \log \frac{\#empty}{size}$

Problem: Inaccurate when the hash table is too full.

ADAPTIVE HASHING

Estan-Varghese



Hash the values $\in [0, 1/2]$ in a table

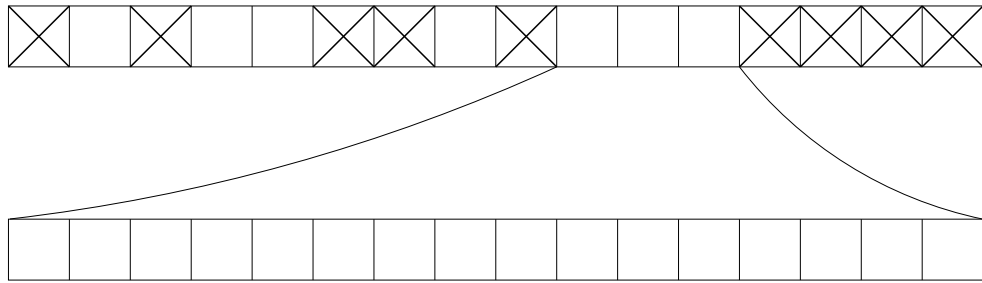
Estimate the number of collisions

Change the zone when the table is over-full.

Standard deviation = $1.5/\sqrt{mem}$ (with assumptions)

SCALABLE MEASUREMENT

Hash the values $\in [0, 1]$ in a table



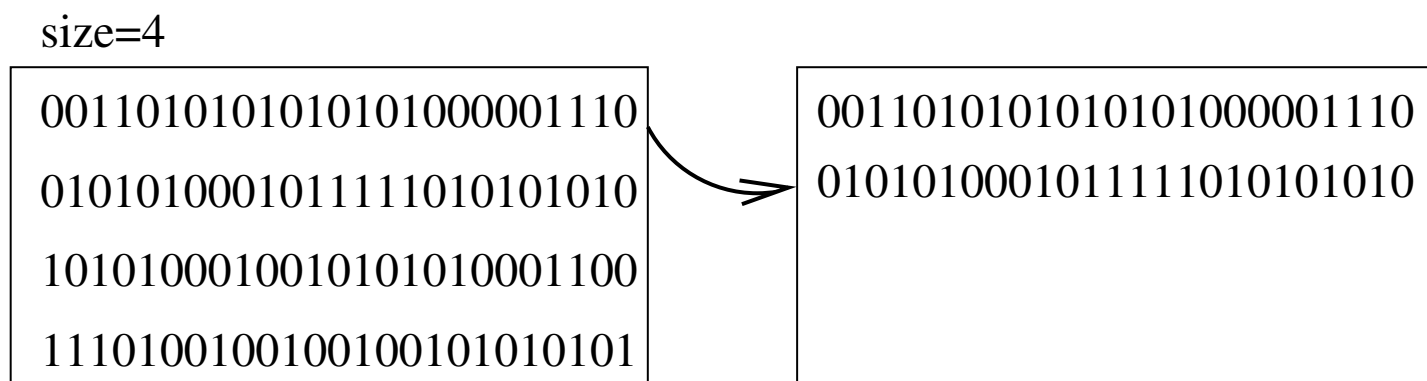
Zoom on an empty zone when the hash table is over-full

Disadvantages:

- Remember the filtered zone
- Too violent zoom

ADAPTIVE SAMPLING

[Flajolet]



- Store up to *size* elements in the bucket,
- Throw away “half” of the bucket + filter, when the bucket is full.

$$\text{standard deviation} = \frac{6.7}{\sqrt{mem}}$$

PROBABILISTIC COUNTING—AN INTUITION

The prefix $\underbrace{00 \dots 00}_k 1$ with k zeros has a probability $\frac{1}{2^{k+1}}$.

Idea

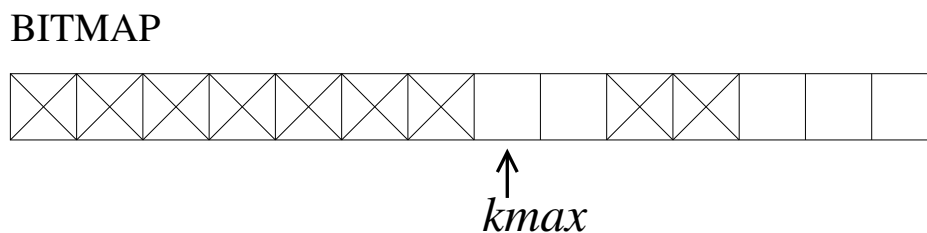
If the prefix of 0's of maximum length has size k , answer 2^k

PROBABILISTIC COUNTING—THE ALGORITHM

[Flajolet-Martin]

The value $\rho(w)$ is the position of the first 1 in the hashed value of w .

Ex: $\rho(00010111) = 4$, $\rho(11000111) = 1$



$$BITMAP[k] = 1 \iff \exists w \text{ s.t. } \rho(w) = k$$

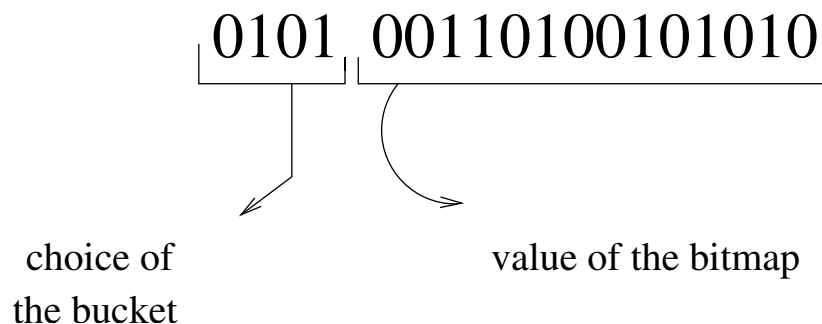
The estimator is $2^{ck_{max}}$

Eventually, use m bitmaps, and m hashing functions.

Problem: cost in time.

PROB. COUNTING WITH STOCHASTIC AVERAGING

- Only one hashing function.
- The items are sent to 2^b buckets according to their b first bits.
- Apply Probabilistic Counting to each bucket with the other bits.

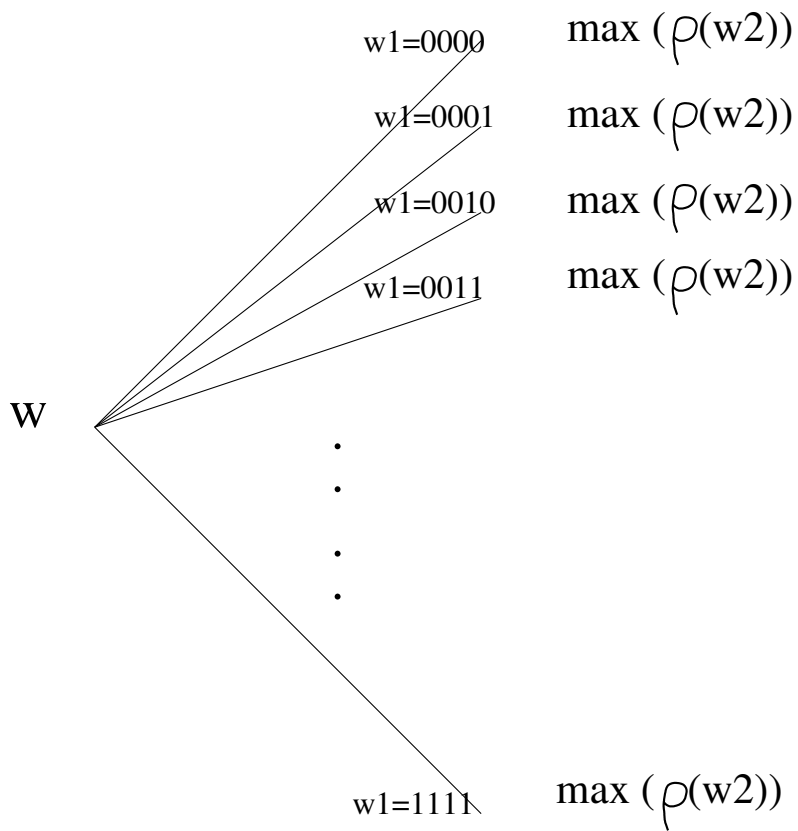


The standard deviation is then $\frac{4.4}{\sqrt{mem}}$

$w =$

0100	001001011110010101
------	--------------------

$w = w_1 \cdot w_2$



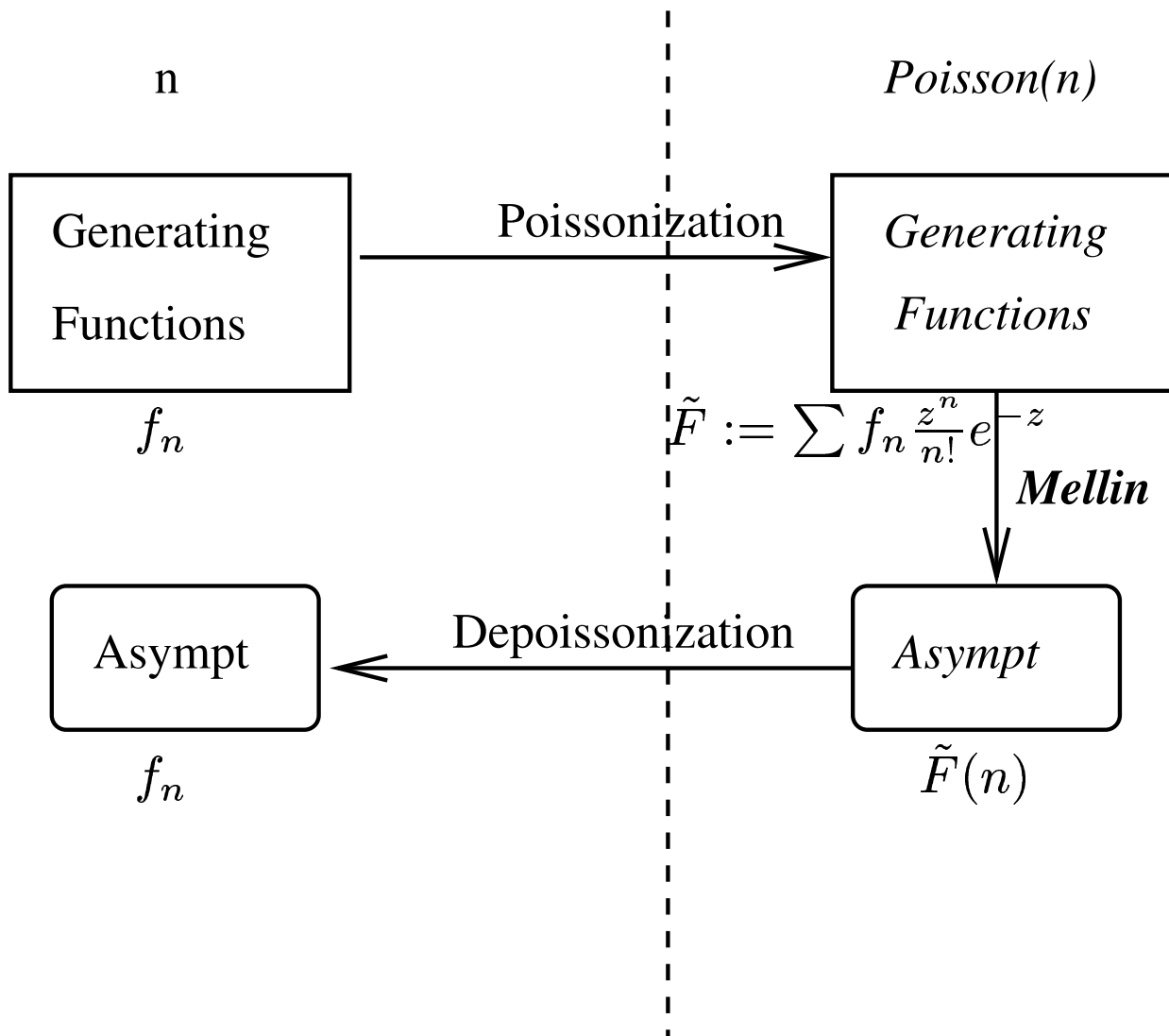
MAXIMUM BASED PROBABILISTIC COUNTING

- $m = 2^b$ buckets
- Each word is sent to a bucket w.r.t. its prefix of length b .
- The bucket B_i contains the max of $\rho(w)$ denoted by X_{n_i} .

The estimator e_n is $m \sum X_{n_i} / m$

- Independent of repetitions
- Memory requirement: $5m$
- Very few calculation
- Experimental standard deviation $2.8 / \sqrt{mem}$

SKETCH OF THE ANALYSIS



ANALYSIS-GENERATING FUNCTIONS

X_n is the random variable corresponding to a bucket filled with n words.

$$\mathbf{P}(X_n = k) = (1 - 1/2^{k+1})^n - (1 - 1/2^k)^n$$

$G(z, u)$ is the probability generating function of X_n

$$G(z, u) = \sum_{n,k} \frac{z^n}{n!} u^k \mathbf{P}(X_n = k).$$

$$P\left(\sum X_i = k\right) = [z^n][u^k] G(z/m, u)^m n!$$

The average and the variance are:

$$E(e_n) = mn![z^n]G(z/m, 2^{1/m})^m$$

$$V(e_n) = m^2 n![z^n]G(z/m, 2^{2/m})^m - (mn![z^n]G(z/m, 2^{1/m})^m)^2$$

THE POISSON MODEL

On Poisson's hint, we study

$$\mathcal{E}_n = mG(n/m, 2^{\frac{1}{m}})^m (e^{-n/m})^m = mA^m$$

$$\mathcal{E}_{2,n} = m^2 G(n/m, 2^{2/m})^m (e^{-n/m})^m = m^2 B^m$$

$$G(z, u) = \sum u^k \left(e^{-1/2^{k+1}} - e^{-1/2^k} \right)$$

$$A(x) = \sum_k 2^{k/m} \left(\psi(x/2^{k+1}) - \psi(x/2^k) \right)$$

with $\psi(x) = e^{-x/m}$.

$A(x)$ is an harmonic function, and $A = A(n)$.

THE MELLIN TRANSFORM

$$A^*(s) = m^s \Gamma(s) (2^s - 1) \frac{1}{1 - 2^{1/m} 2^s}$$

- Singularity at $s = \frac{-1}{m}$
- The residue is $a = m^{-1/m} \Gamma(-1/m) (2^{-1/m} - 1) \log 2$
- By Mellin's reverse mapping $A \sim an^{1/m}$

With the same technique we obtain $B \sim bn^{2/m}$

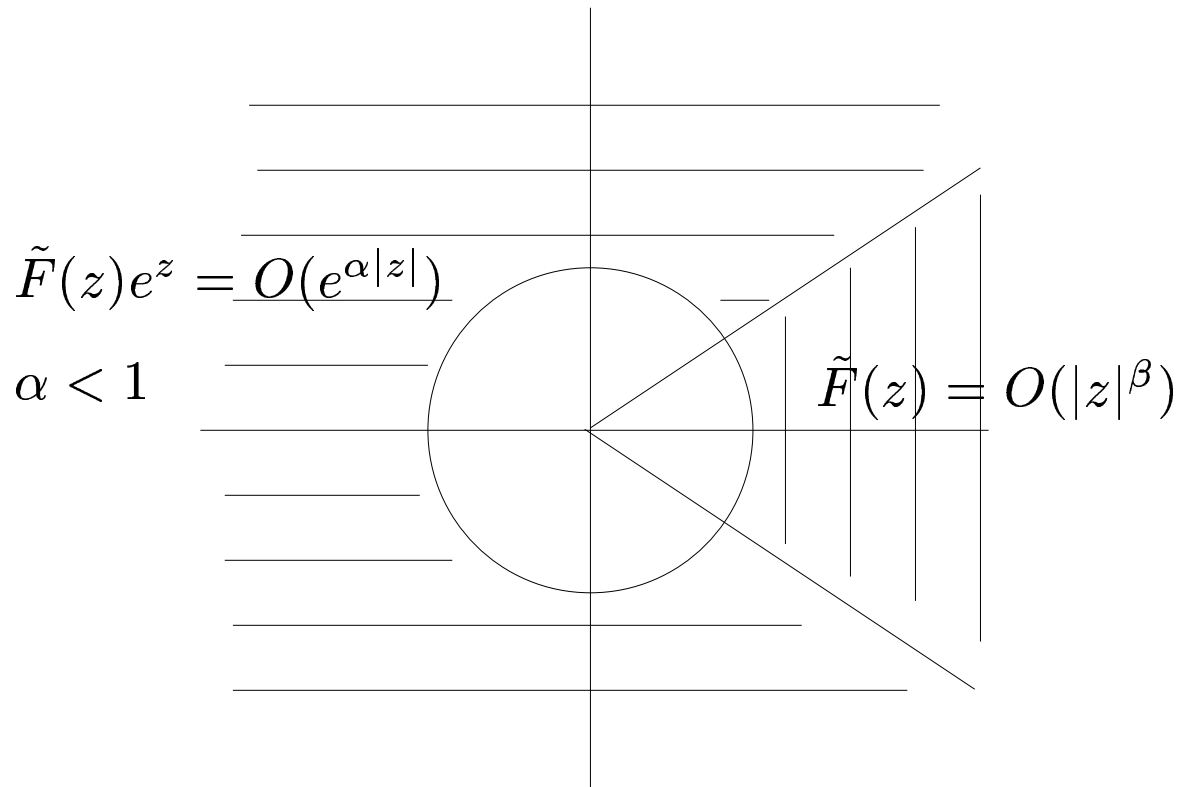
The standard deviation is

$$\sigma^2 = \frac{m^2 B^m}{(mA^m)^2} - 1 = \frac{1}{m} \left(\frac{\pi^2}{6} + \frac{\log^2 2}{12} + o(1) \right)$$

$$\sigma \sim \frac{2.90}{\sqrt{mem}}$$

DEPOISSONIZATION

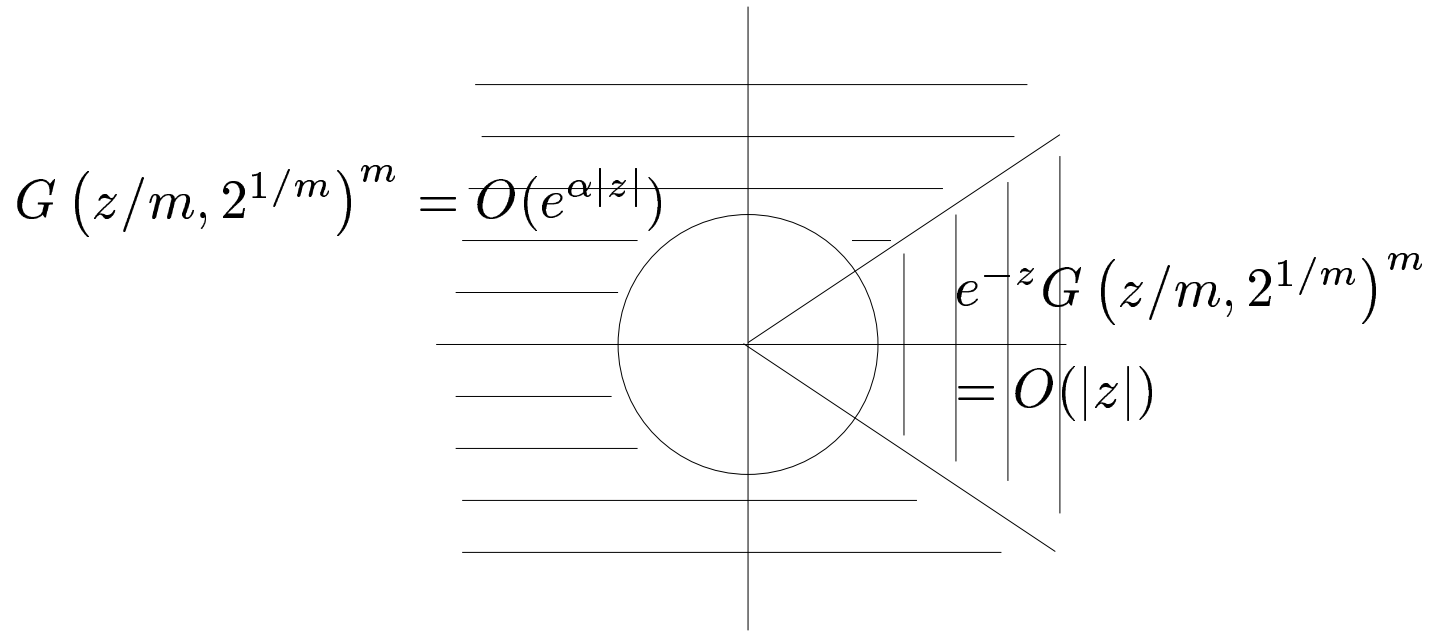
[Basic Depoissonization lemma–Jacquet-Szpankowski]



Then

$$f_n = \tilde{F}(n) + O(n^{\beta-1})$$

DEPOISSONIZATION (2)



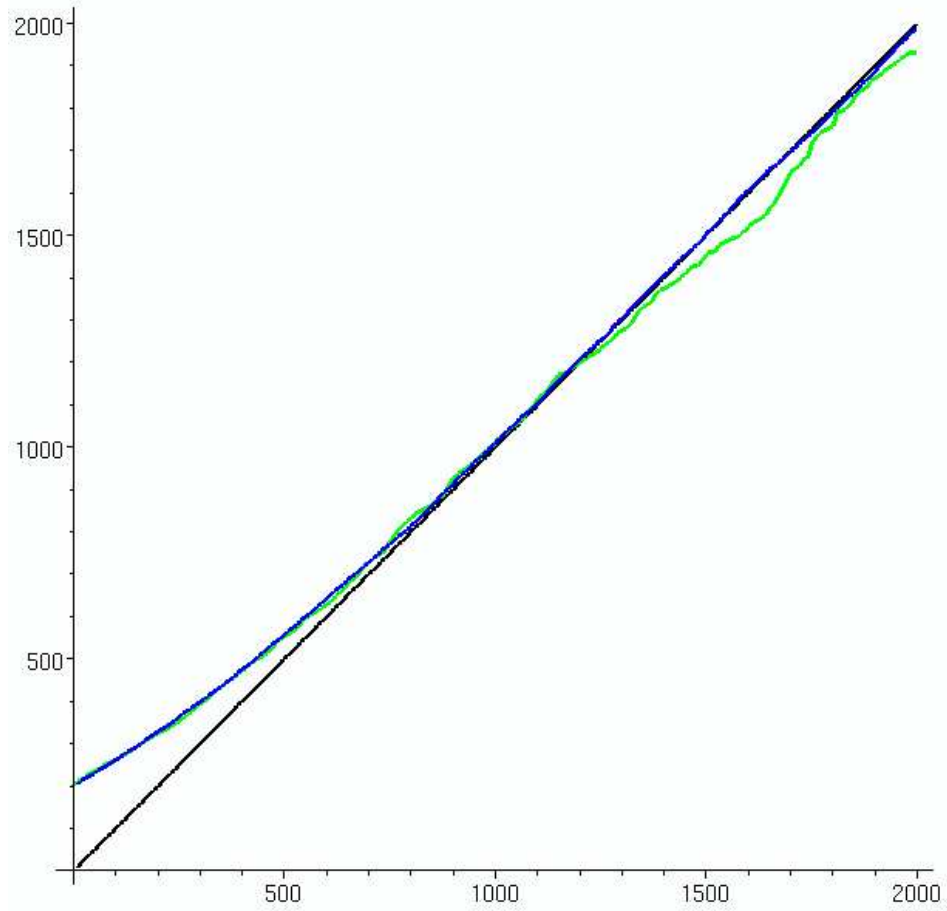
Then

$$\mathcal{E}_n \sim E(e_n)$$

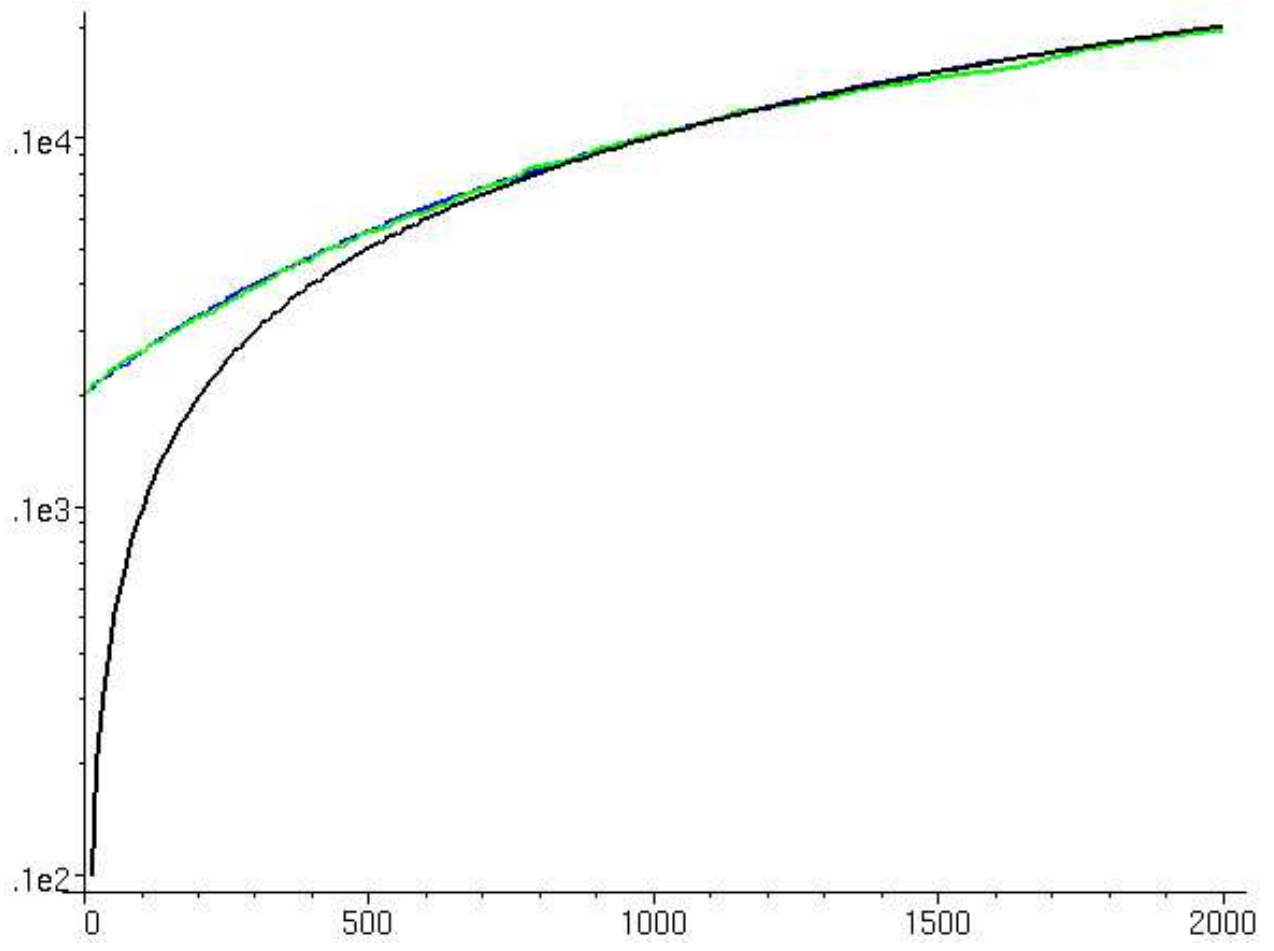
Similarly

$$\mathcal{E}_{2,n} \sim E(e_n^2)$$

EXPERIMENTAL RESULTS

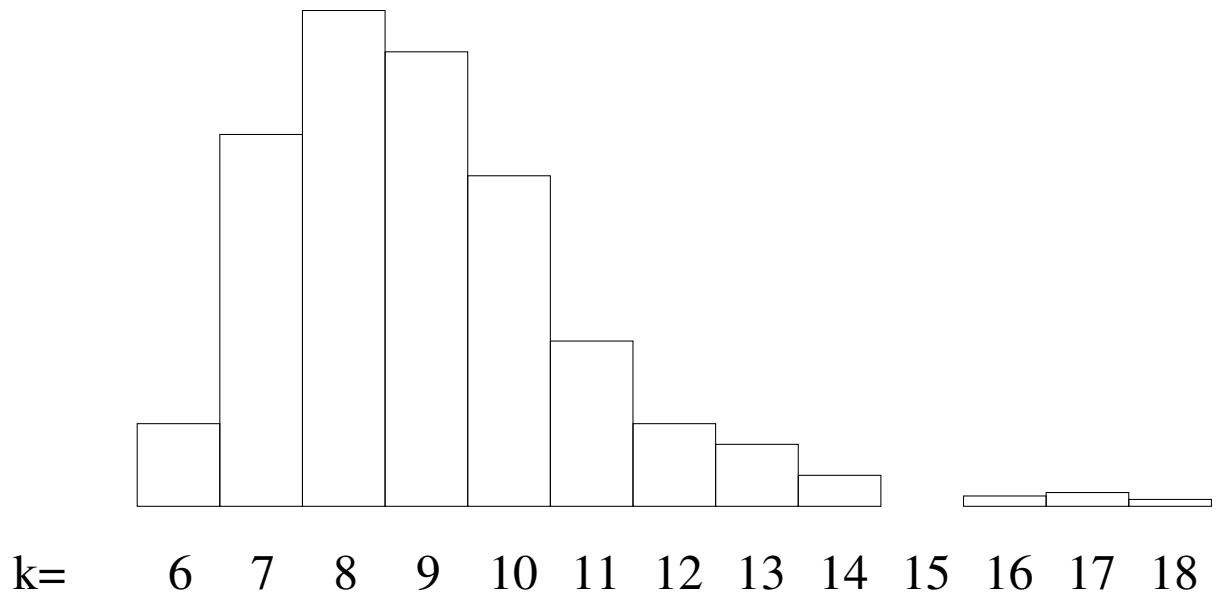


$m=512$, one simulation



The same simulation, on logarithmic scale

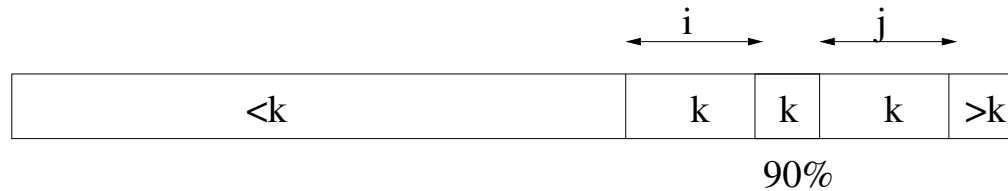
THE VALUES OF THE BUCKETS



Values of the buckets for $n = 100000$ and $m=512$

REDUCING THE STANDARD DEVIATION

IDEA : truncate the non-meaningful values.



$$\sum_k \sum_i \sum_j ([u^k]G(z/m, uv))^{i+1} ([u^k]G(z/m, u))^j \left(G(z/m, 1) - [u^k] \frac{G(z/m, u)}{1-u}\right)^{\frac{m}{10} - j} \left([u^{k-1}] \frac{G(z/m, uv)}{1-u}\right)^{\frac{90m}{100} - i - 1} \binom{m}{\frac{90m}{100} - i - 1, i + 1 + j, \frac{m}{10} - j}$$

$$\sigma = 2.8 / \sqrt{mem}$$

$$\sigma = 2.4 / \sqrt{mem} \text{ when } 90\% \text{ truncated}$$

$$\sigma = 2.2 / \sqrt{mem} \text{ when } 80\% \text{ truncated}$$

$$\sigma = 2.5 / \sqrt{mem} \text{ when } 70\% \text{ truncated}$$

Idea : use a least-square regression

OTHER RELATED PROBLEMS

$$m_i = \#\{elements = i\}$$

- Counting the number of elements

$$\sum m_i$$

Morris

- Measuring the repetitions

$$\sum m_i^2$$

Alon-Matias-Szegedy

$$\sum m_i^k$$