Séminaire COLLÈGE DE FRANCE 1530 chaire Prof. Gérard Berry 

Algorithmes probabilistes pour de grandes masses de données

Philippe Flajolet,
Algorithms; INRIA–Rocquencourt

— Le 25 janvier 2008 —

In *“Pourquoi et comment le monde devient numérique”*

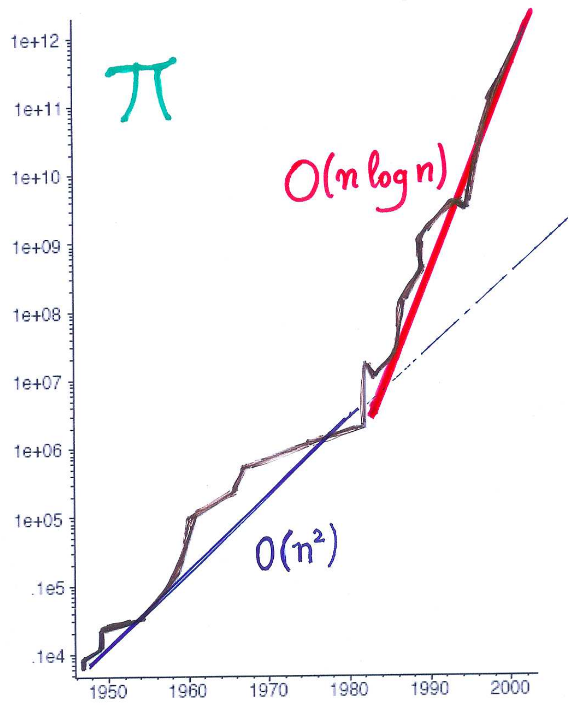
1 LES ALGORITHMES, CŒUR DE L'INFO...

"La Loi de Moore tue l'algorithmique!" (proverbe populaire)

- Calcul de Pi:

ENIAC 1949: $\dfrac{1120\,\mathrm{D}}{1000\,\mathrm{I/S}}$; Kanada 2002: $\dfrac{2 \cdot 10^{12}\,\mathrm{D}}{10^{12}\,\mathrm{I/S}}$.

"*La Loi de Moore tue l'algorithmique!*" (proverbe populaire)



ENIAC 1949: $\dfrac{1120\,\mathrm{D}}{1000\,\mathrm{I}/S}$;

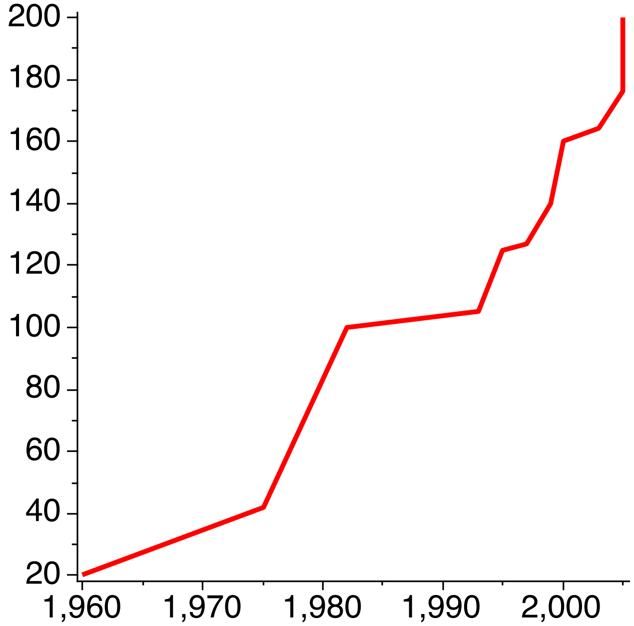
Kanada 2002: $\dfrac{2 \cdot 10^{12}\,\mathrm{D}}{10^{12}\,\mathrm{I}/S}$.

FFT (Fourier) + AGM + hypergeométriques

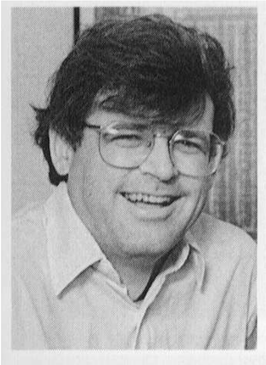
(BBP: *"The quadrillionth bit of π is* **0**.")

- Factorisation d'entiers:

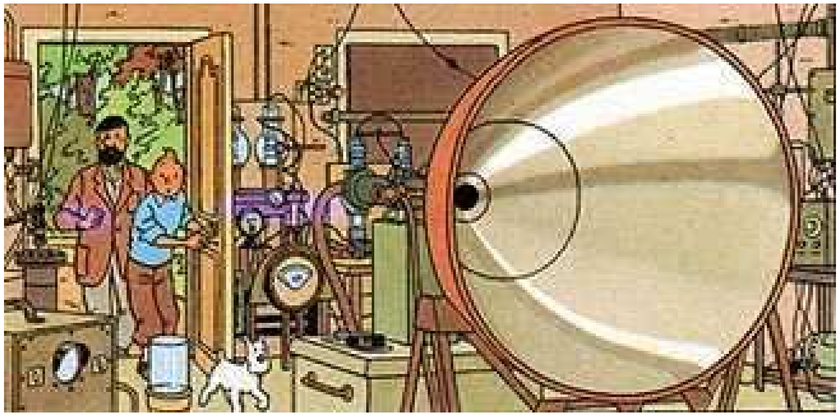
$$10^{30} \text{ en } 1965; \qquad 10^{200} \text{ en } 2005.$$



$\rightsquigarrow$ cryptographie RSA



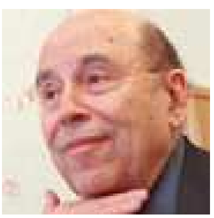
SEDGEWICK'S PRINCIPLE: *"Volumes and complexity of data increase faster than processing speed. We need* **ever better algorithms** *to keep pace."*



Critères algorithmiques:

• Pire cas (!)

Révolution Knuth (1970):
Parier sur les données “typiques”.



Révolution Rabin (1980):
Introduire volontairement l’aléa dans le calcul.

$\leadsto$ Modèles et analyses *mathématiques*.

EXEMPLE: le hachage

Stocker x a l'adresse $h(x)$.

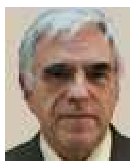
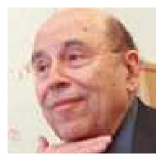
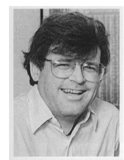
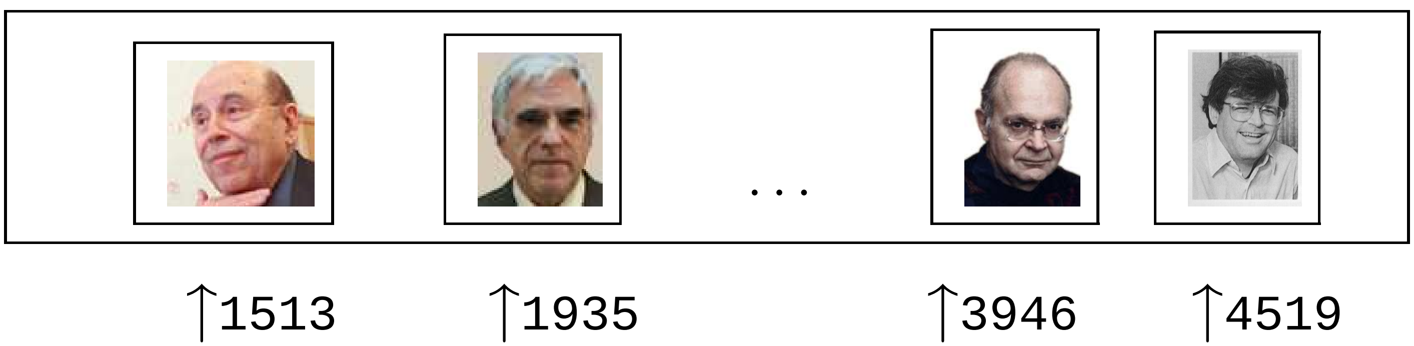
Fichier de , , ,  ...

TABLE = 

— Le choix d'une "bonne" fonction donne un *pseudo-aléa*.

— Probabilités classiques: *allocations aléatoires* n (objets) $\mapsto$ m (cases)

$$\text{Loi de Poisson:} \quad \mathbb{P}(C = k) \sim e^{-\lambda} \frac{\lambda^k}{k!}; \qquad \lambda := \frac{n}{m}.$$

— Gestion de collisions: $\rightsquigarrow$ combinatoire analytique

$$\text{equation fonctionnelle:} \qquad \frac{\partial F(z,q)}{\partial z} = F(z,q) \cdot \frac{F(qz,q) - qF(z,q)}{q-1}.$$

(Knuth 1965; Knuth 1998; F-Poblete-Viola 1998; F-Sedgewick 2008)

2 ALGORITHMIQUE DES FLUX MASSIFS

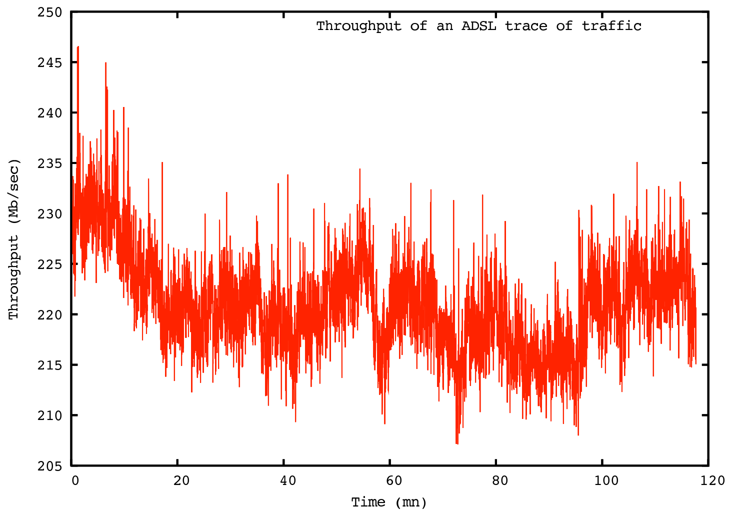

Routeurs $\approx$ Terabits/sec (10^{12}b/s).

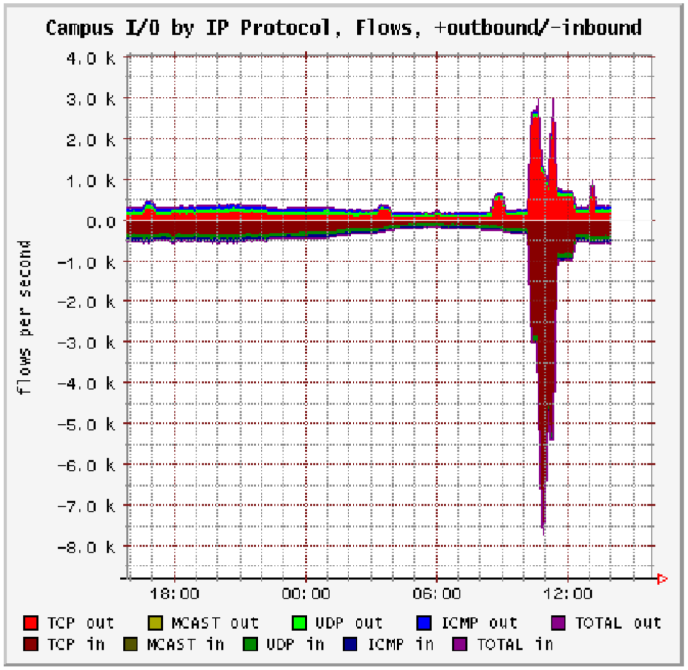
Google indexe 10 milliards de pages & prepare 100 Petabytes de données (10^{17}B).

Algorithmes de flux (*stream algorithms*)
= une passe; mémoire $\leq$ une page A4

Cf. Leçon inaugurale.

Exemple: Propagation de virus et détection d'attaques sur réseaux





(Trafic ADSL brut)

Volume brut

(Attaque)

Cardinalité

Exemple: Le problème de cardinalité

— Donnée: flux $s = s_1 s_2 \cdots s_\ell, \qquad s_j \in \mathcal{D}, \quad \ell \propto 10^9$.

— Sortie: Estimation de la cardinalité $n, \quad n \propto 10^7$.

— Conditions:

très peu de mémoire auxiliaire;
une seule passe "simple";
aucune hypothèse statistique.
précision à 1 ou 2% près.

Plus généralement. . .

- Cardinalité: nombre de valeurs distinctes;
- Icebergs: nombre d'éléments de fréquence relative $> 1/30$;
- Souris: nombre d'éléments de fréquence absolue < 10;
- Éléphants: nombre d'éléments de fréquence absolue > 100;
- Moments: mesure de "profil" des données. . .

Applications: réseaux; fouille *quantitative* de données; grandes bases de données et esquisses; internet; analyse statistique rapide de séquences.

3 ICEBERGS

Un k-*iceberg* est une valeur de fréquence relative $> 1/k$.

`abracadabraba babies babble bubbles alhambra`

très peu de mémoire auxiliaire;
une seule passe "simple";
aucune hypothèse statistique;
précision à 1 ou 2% près.

$k = 2$. Majorité = 2-iceberg: `a b r a c a d a b r a ...`

La guerre des gangs $\equiv$ 1 registre $\langle$`valeur,compteur`$\rangle$

$k > 2$. Généralisation avec $k - 1$ registres.

Fournit un surensemble sans perte des icebergs.
(Filter en combinant avec échantillonnage.)

(Karp-Shenker-Papadimitriou 2003)

4 CARDINALITÉ

- Le hachage fournit des valeurs (quasi) aléatoires uniformes.
- L'aléa obtenu est reproductible:

provence	poitou	berry	$\cdots$	poitou	$\cdots$
	3589			3589	

Le flux de données = un multi-ensemble $\leadsto$ réels uniformes $[0,1]$
Une $\boxed{\text{observable}}$ = une fonction de l'ensemble sous-jacent.

Une $\boxed{\text{observable}}$ = une fonction de l'ensemble haché.

— A. On a vu passer le motif initial `0.011101`

— B. Le minimum des valeurs lues est `0.0000001101001`

— C. On a vu les motifs initiaux `0.1...` *et* `0.01...` *et* `0.001...`

— D. On a vu passer tous les motifs `0.`$x_1 \cdots x_{20}$ pour $x_j \in \{0, 1\}$.

NB: "On a vu passer 1968 `bits = 1` n'est pas une observable.

Vraisemblablement(??):

A indique $n > 2^6$; B indique $n > 2^7$; C indique $n \geq 8$; D indique $n \geq 2^{20}$.

4.1 Hyperloglog

Les rouages du meilleur algorithme connu

Étape 1. Choisir l'observable.

L'observable O vaut le maximum des positions du premier 1

11000	10011	00010	10011	01000	00001	01111
1	1	2	1	2	**5**	2

= **un seul registre numérique** < 32 $(n < 10^9)$

$\equiv$ **un "petit" octet** (5 bits)

(F-Martin 1985); (Durand-F. 2003); (F-Fusy-Gandouet-Meunier 2007)

Étape 2. Analyser l'observable.

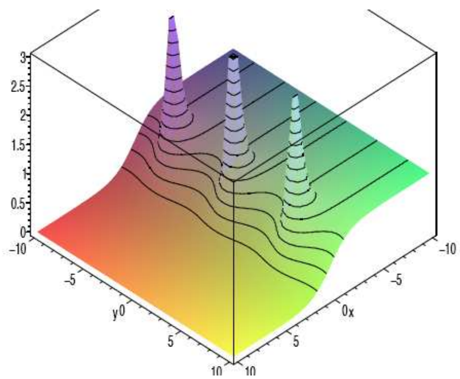
Theorème.
(i) Espérance: $\mathbb{E}_n(O) = \log_2(\varphi n) + \text{oscillations} + o(1)$.
(ii) Variance: $\mathbb{E}_n(O) = \xi + \text{oscillations} + o(1)$.

On *estime* la valeur logarithmique de n avec un biais systématique (φ) et une dispersion (ξ) de $\pm$ 1 ordre binaire de grandeur.

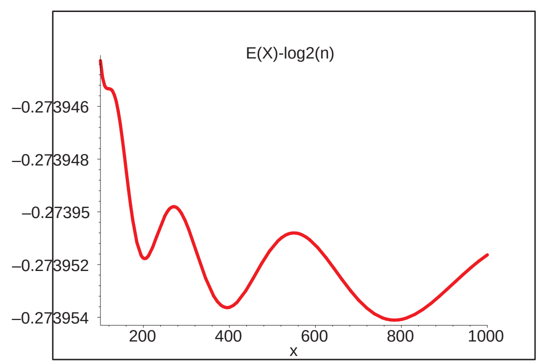
$\rightsquigarrow$ corriger le biais; améliorer la précision!

La transformation de Mellin: $\int_0^\infty f(x)x^{s-1}\,dx$.

- *Factorise les superpositions linéaires de modèles à différentes échelles.*
- *Relie singularités complexes de* $\int$ et asymptotique.



(singularités)



(asymptotique)

Algorithme Squelette(S : flux):
initialiser un registre $R := 0$;
pour $x \in S$ **faire**
$h(x) = b_1 b_2 b_3 \cdots$;
$\rho := \text{position}_{1\uparrow}(b_1 b_2 \cdots)$;
$R := \max(R, \rho)$;
calculer l'estimateur de $\log_2 n$.

= un seul "petit octet" de $\boxed{\log_2 \log_2 N}$ bits: 5 bits pour $N = 10^9$;

= correction de l'estimateur par $\varphi = e^{-\gamma}/\sqrt{2}$; (constante d'Euler)

= non biaisé; précision limitée: $\pm$ un ordre de grandeur binaire.

Étape 3. Fabriquer un véritable algorithme.

Plan A: Répéter m fois l'expérience & prendre la moyenne.
+Corriger le biais.

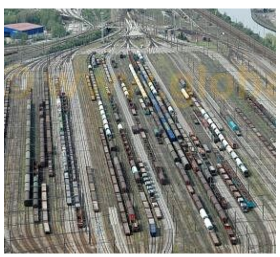
Estime $\log_2 n$ avec precision $\approx \pm \dfrac{1}{\sqrt{m}}$.

($m = 1000 \Longrightarrow$ précision = quelques pourcents.)

Coût de calcul multiplié par m.

Imprécision due à la dépendance..

Plan B ("Stochastic averaging"): On divise les données en m lots; on calcule une moyenne des estimations des lots.



Algorithme HyperLoglog(S : flux; $m = 2^{10}$):
initialiser m registres $R[\,]$:= 0;
pour $x \in S$ **faire**
$\quad h(x) = b_1 b_2 \cdots; \quad A := \langle b_1 \cdots b_{10} \rangle_{\text{base } 2};$
$\quad \rho := \text{position}_{1\uparrow}(b_{11} b_{12} \cdots);$
$\quad R[A] := \max(R[A], \rho);$
calculer l'estimateur de la cardinalité n.

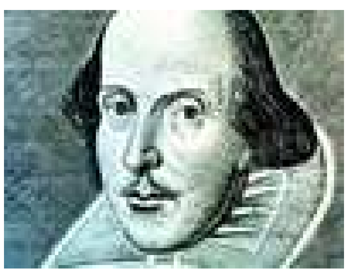
L'algorithme complet comprend $O(12)$ instructions + hachage.
Il calcule la moyenne *harmonique* des $2^{R[j]}$; puis multiplie par m.
Il corrige le biais systématique; enfin le biais non asymptotique.

L'analyse mathématique (combinatoire, probabiliste, asymptotique) intervient de manière non triviale dans la conception.
(ici: Mellin + méthodes de col).

$\rightsquigarrow$ Pour m registres, l'erreur standard est en $\frac{1.035}{\sqrt{m}}$.

Avec 1024 octets, on estime les cardinalités jusqu'à 10^9 avec une erreur standard de 1.5%.

Whole of Shakespeare: 128bytes $(m = 256)$



$\mapsto$

```
ghfffghfghgghggggghghheehfhfhhgghghghhfgffffhhhiigfhhffgfiihfhhh
igigighfgihfffghigihghigfhhgeegeghgghhhgghhfhidiigihighihehhhfgg
hfgighigffghdieghhhgghhfghhfiiheffghghihifgggffihgihfggighgiiif
fjgfgjhhjiifhjgehgghfhhfhjhiggghghihigghhihihgiighgfhlgjfgjjjmfl
```

Estimate $n^\circ \approx 30,897$ against $n = 28,239$ distinct words.

Error is $+9.4\%$ for **128 bytes**(!!)

4.2 Applications distribuées

Étant donnés 90 annuaires, combien de noms?

Collection des registres $R_1, \ldots, R_m$ de $S \equiv$ signature de S.

Signature d'une union $\equiv$ max/composantes ($\vee$):

$$\begin{cases} \mathbf{sign}(A \cup B) & = & \mathbf{sign}(A) \vee \mathbf{sign}(B) \\ |A \cup B| & = & \mathbf{estim}(\mathbf{sign}(A \cup B)). \end{cases}$$

On peut estimer à 1% près le nombre de noms en envoyant 89 fax, chacuns d'un quart de page A4 env.

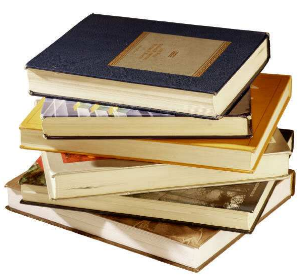
4.3 Comparaison de documents

Pour S un flux (sequence, multi-ensemble):

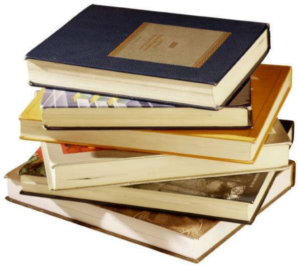
- la taille $\|S\|$ = nombre total d'éléments;
- la cardinalité $|S|$ = nombre d'éléments distincts.

Pour A, B deux flux, l'indice de similarité (Broder 1997–2000) est

$$\mathbf{simil}(A, B) := \frac{|A \cap B|}{|A \cup B|} \equiv \frac{\text{vocabulaire commun}}{\text{vocabulaire total}}.$$



Peut-on classer un million de livres, selon leur similarité, avec un ordinateur portable?



Peut-on classer 1.000.000 millions de livres, selon leur similarité, avec un ordinateur portable?

$$\begin{cases} |A| & = & \mathbf{estim}(\mathbf{sign}(A)) \\ |B| & = & \mathbf{estim}(\mathbf{sign}(B)) \\ |A \cup B| & = & \mathbf{estim}(\mathbf{sign}(A) \vee \mathbf{sign}(B)) \end{cases} \qquad \mathbf{simil}(A, B) = \frac{|A| + |B| - |A \cup B|}{|A \cup B|}.$$

Soit une bibliothéque de N livres (e.g.: $N = 10^6$) ayant un volume total de V caractères (e.g.: $V = 10^{11}$).

— Solution exacte: coût temps $\simeq N \times V$.
— Solution par signatures: coût temps $\simeq V + N^2$.

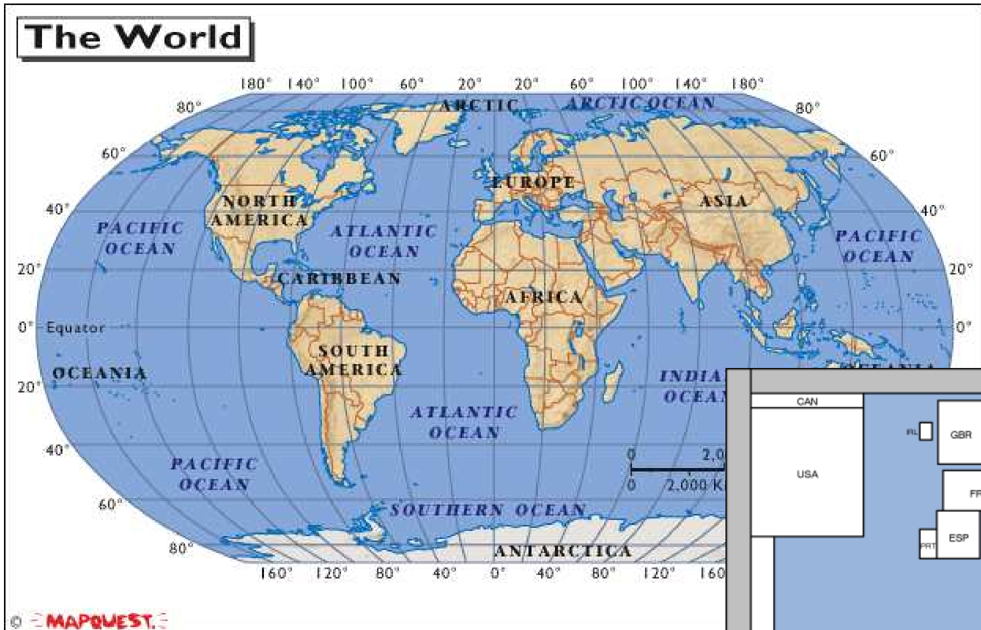
Match: signatures = $\boxed{10^{12}}$ contre exact = $\boxed{10^{17}}$.

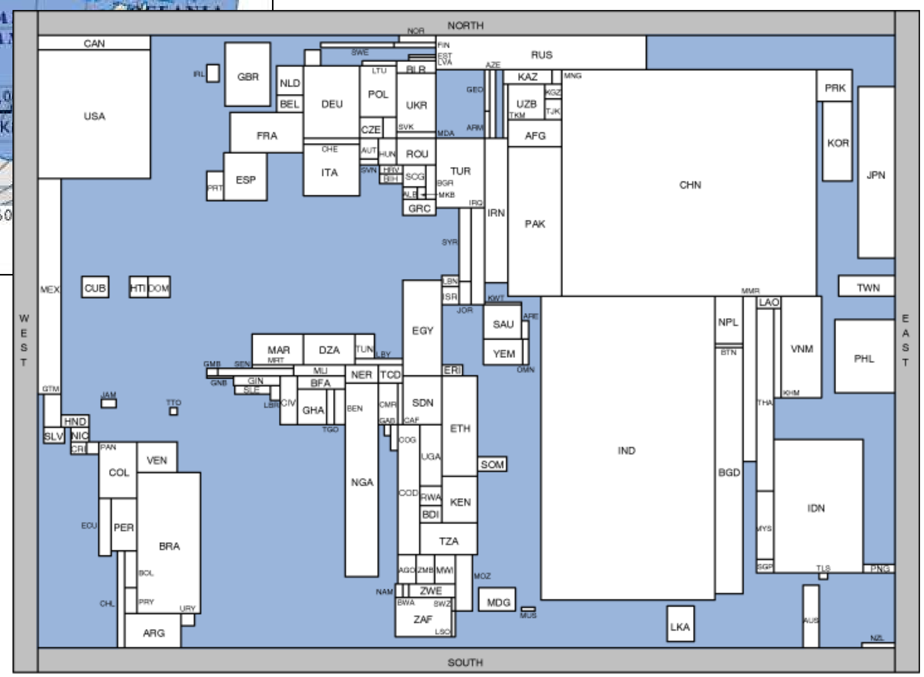
5 ÉCHANTILLONNAGE ADAPTATIF

Peut-on localiser le centre de gravité géographique de la France; donnée: 60 millions de $\langle$`personnes & communes`$\rangle$ *?*

— Exact: oui bien sûr = élimination des doublons (“projection”)

— Approché: échantillonnage simple $\Longrightarrow$ au Sud Est de Paris(!!).

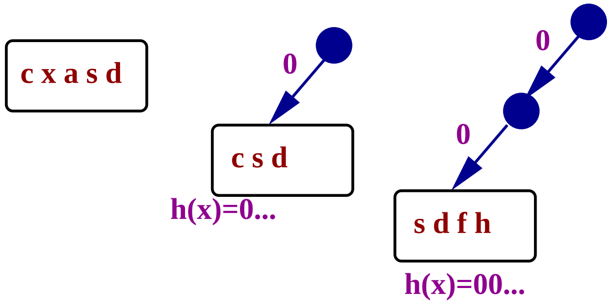




©Bettina Speckmann, TU Eindhoven)

Échantillonner sur le domaine des valeurs ***distinctes****?*

L´echantillonnage adaptatif:



Algorithme: Adaptive Sampling(S : flux);

$C := \emptyset$;{cache de capacité m}

$p := 0$; {profondeur}

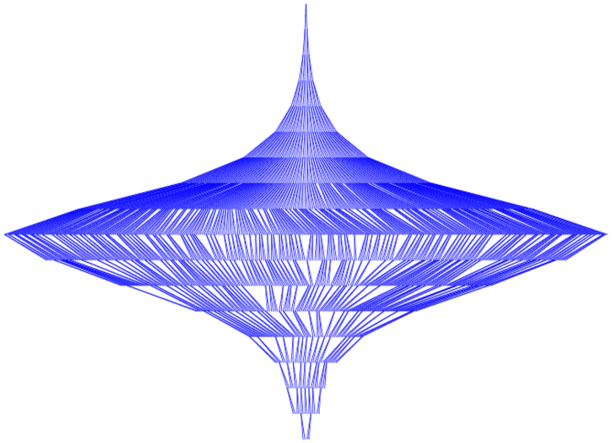
pour $x \in S$ **faire**

  si $h(x) = 0^p \cdots$ **alors** $C := C \cup \{x\}$;

  si déborde(C) **alors** p := p+1; filtrer C;

retourner C {$\approx m/2 \ldots m$ éléments}.

(Wegman 1980) (F 1990) (Louchard 97)



L'**analyse** se relie à la structure d'arbre digital: compression de données; recherche textuelle; protocoles de communication; &c.

- Donne un échantillon non-biaisé de **valeurs distinctes**;
- Donne un nouvel algorithme non biaisé de **cardinalité**:

$$\mathrm{estim}(S) := |C| \cdot 2^p.$$

Hamlet

• **Échantillonnage simple** (13 éléments):

and, and, be, both, i, in, is, leaue, my, no, ophe, state, the

$\boxed{\texttt{Google}}$ (leaue$\mapsto$leave, ophe$\mapsto$ $\emptyset$) = $\boxed{\texttt{38,700,000}}$.

• **Échantillonnage adaptatif** (10 elements):

danskers, distract, fine, fra, immediately, loses, martiall, organe, passeth, pendant

$\boxed{\texttt{Google} = 8}$, tous vers Shakespeare/ Hamlet $\leadsto$ *mice,* later!

6 LES SOURIS

Échantillonnage adaptatif plus compteurs!

— Hamlet: *danskers*^{1}, *distract*^{1}, *fine*^{9}, *fra*^{1}, *immediately*^{1}, *loses*^{1}, *martiall*^{1}, *organe*^{1}, *passeth*^{1}, *pendant*^{1}.

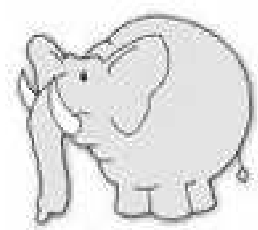
Cache de taille =100, donne un échantillon de 79 éléments.

$$\mathbf{1^{50}, 2^{14}, 3^{4}, 4^{2}, 5^{1}, 6^{1}, 9^{1}, 13^{1}, 15^{1}, 28^{1}, 43^{2}, 128^{1}}.$$

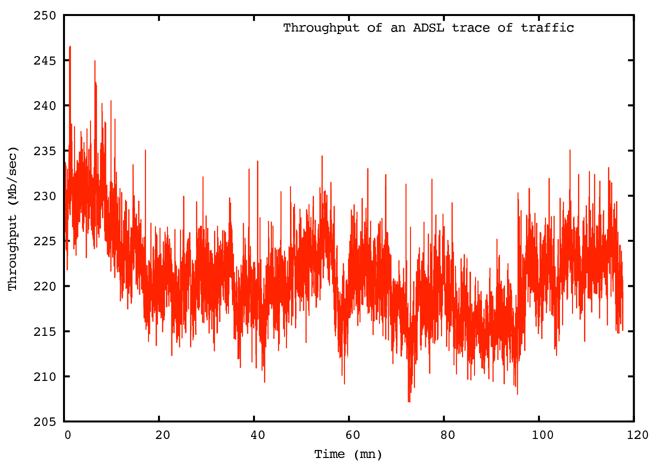
	1-Souris	2-Souris	3-Souris
Estimé	63%	17%	5%
Réel	60%	14%	6%

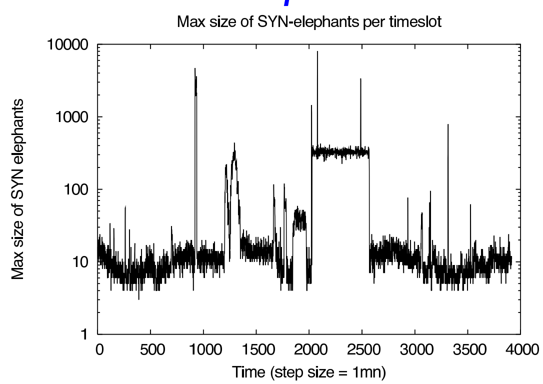
Les 10 mots les plus fréquents dans Hamlet sont: *the, and, to, of, i, you, a, my, it, in*. Ils représentent $>$ 20% du texte. Avec 20 mots, on capture 30%; avec 50 mots, 44%. **70 mots capturent 50% du texte!**.

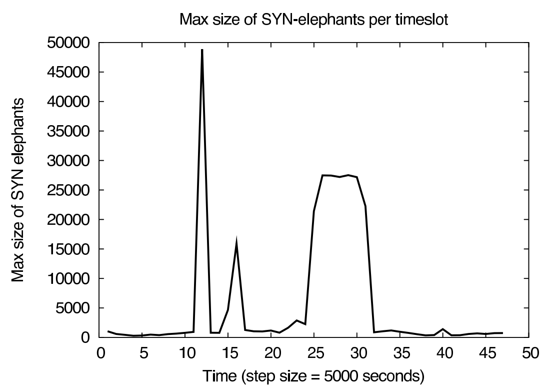
7 LES ÉLÉPHANTS



Un k-éléphant est une valeur dont la fréquence absolue est $\geq k$.





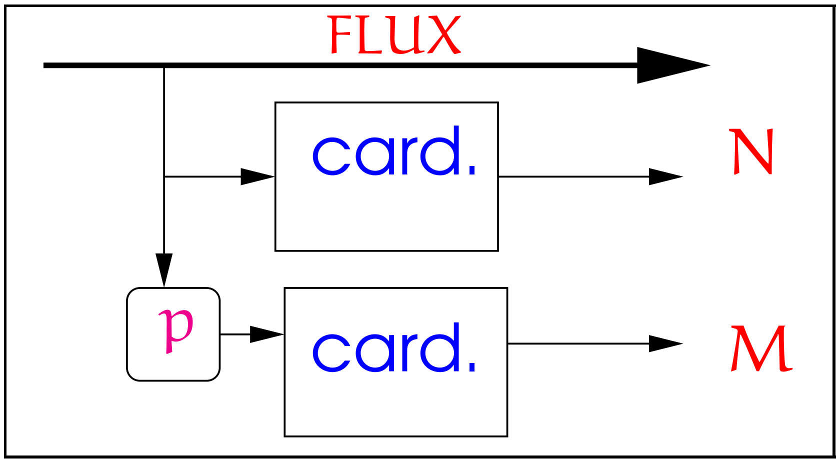


Attaques réseaux par déni de service (Y. Chabchoub, Ph. Robert)

Theorème de complexité (Alon *et al.*) *On ne peut pas déterminer la plus grande fréquence avec une mémoire sous-linéaire.*

- On ne peut pas trouver (miraculeusement) une aiguille dans une botte de foin.
- Mais on peut y trouver (assez facilement) beaucoup d'informations!

Trafic bi-modal: Un flux composé de 1-souris et 10-éléphants

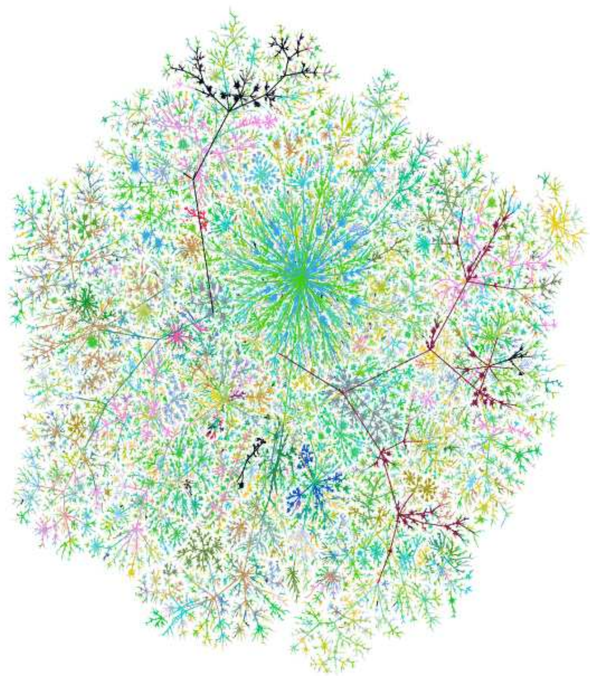


$$\begin{cases} & (p = \frac{1}{10}) \\ N = & N_s + N_e + \text{bruit} \\ M = & \frac{1}{10} N_s + 0.65 N_e + \text{bruit} \end{cases}$$

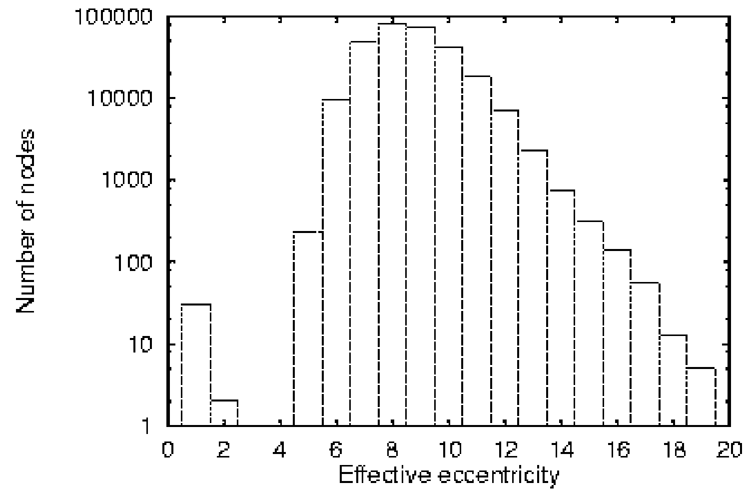
Solution: $\boxed{N_e \approx \frac{10M - N}{5.5}}$ (A. Jean-Marie, O. Gandouet, 2007)

• Nombre de liaisons bi-directionnelles dans un grand graphe; nombre de triangles.

• L'histogramme d'excentricité dans le graphe de l'internet



$\Longrightarrow$

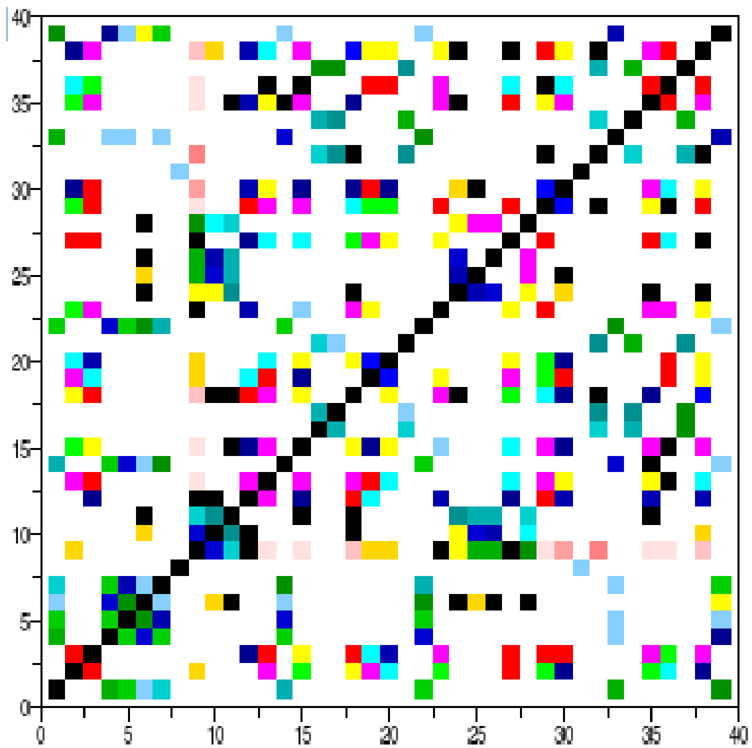


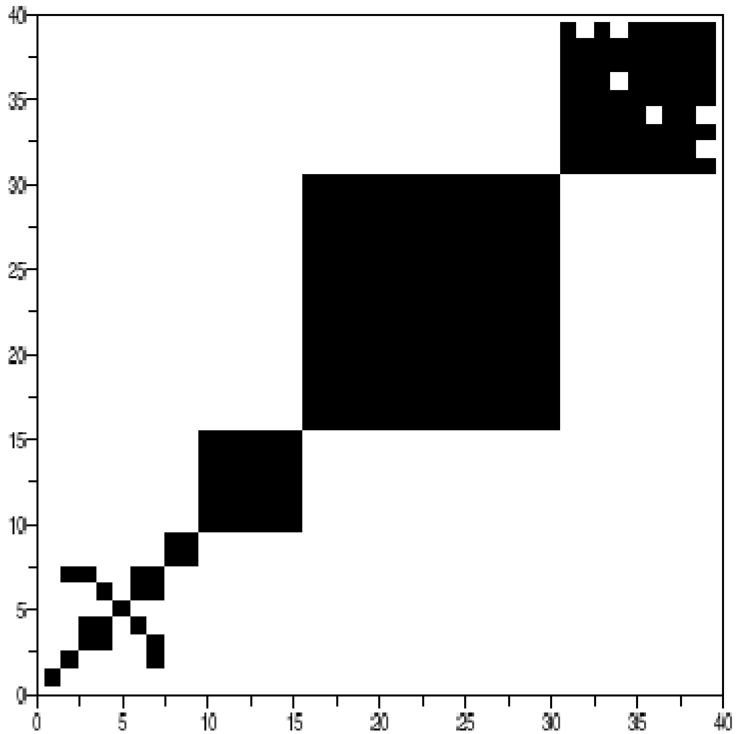
b) Histogram of diameters

Gain: $\times 300$.

(Palmer, Gibbons, Faloutsos^2, Siganos 2001) Internet graph: 285k nodes, 430kedges.

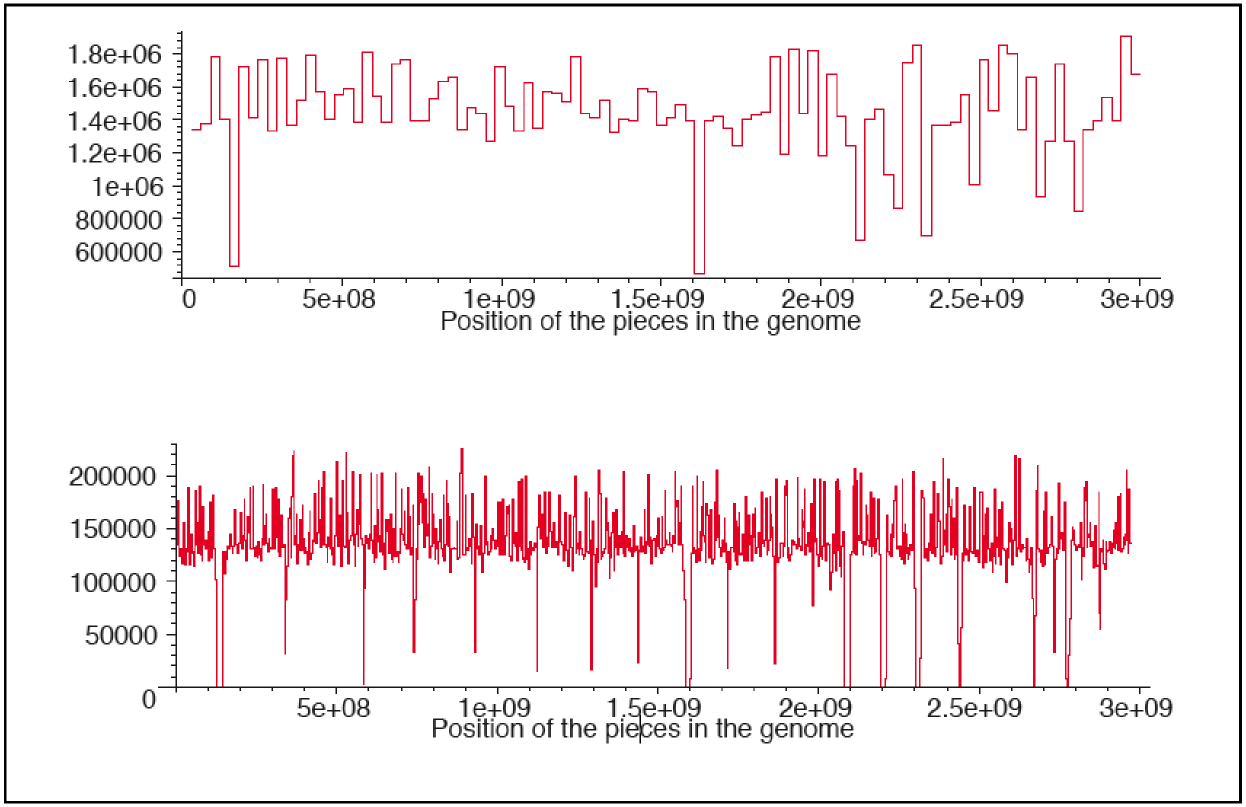
Combien de langues?





(Pranav Kashyap: word-level encrypted texts; classification by language; use $\vartheta = 20\%$ sim.)

Génome

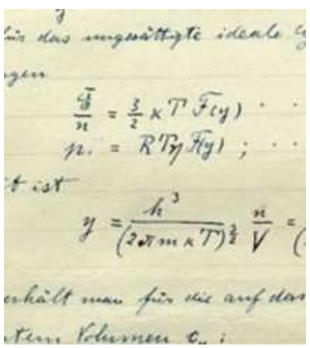


(Giroire 2006: # patterns of length 13 in genome)

Conclusions (?)

Interprétation des résultats = un autre métier.

Possibilités (avec limites!) de l'algorithmique probabiliste.



Continuum: sciences $\rightsquigarrow$ informatique $\rightsquigarrow$ technologie.