

Séminaire



COLLÈGE
DE FRANCE
—1530—

chaire Prof. Gérard Berry



Algorithmes probabilistes pour de grandes masses de données

Philippe Flajolet,
Algorithms; INRIA–Rocquencourt

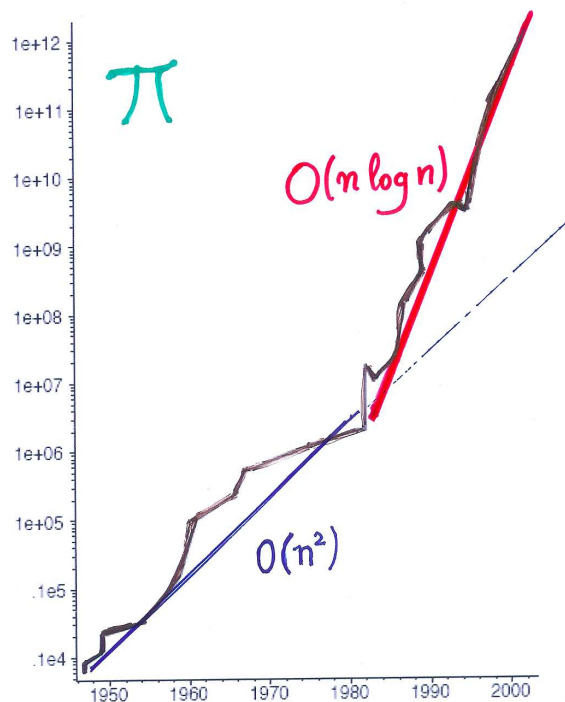
— Le 25 janvier 2008 —

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:



FFT (Fourier) + AGM + hypergéométriques

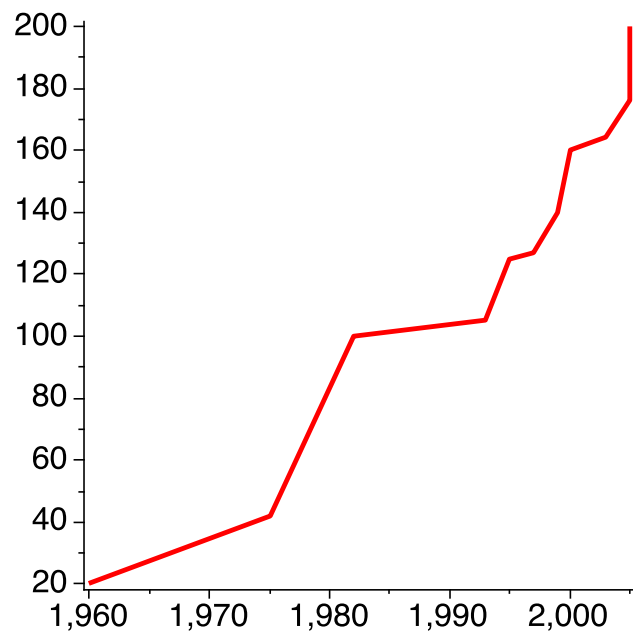
ENIAC 1949: $\frac{1120 D}{1000 I/S}$;

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

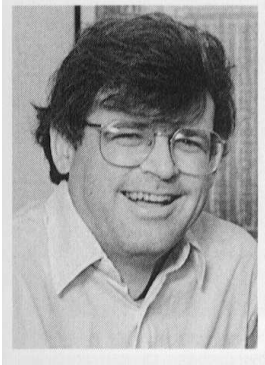
(BBP: *"The quadrillionth bit of π is **0**."*)

- Factorisation d'entiers:

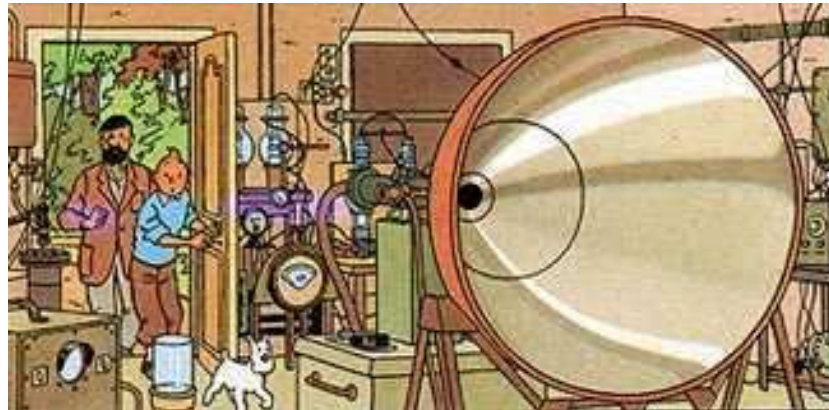
10^{30} en 1965; 10^{200} en 2005.



~> 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):

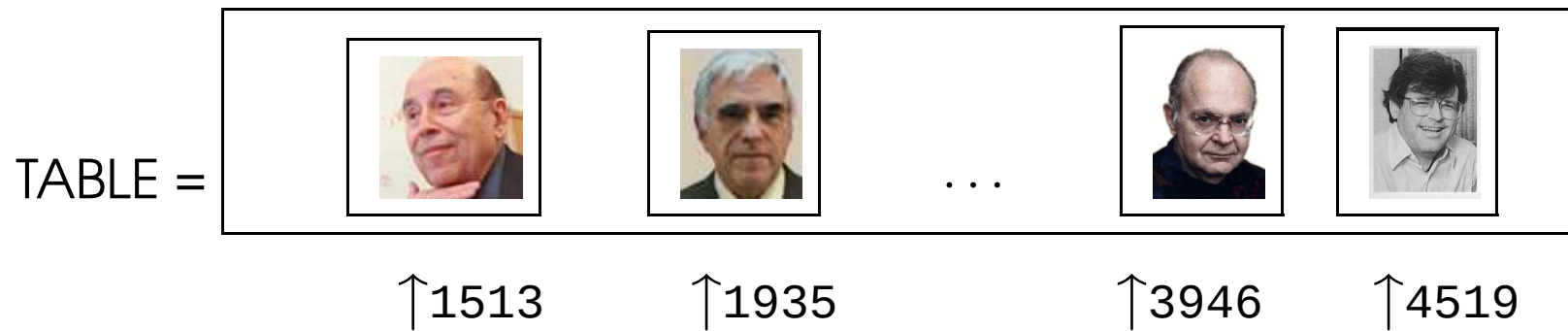
Introduire volontairement l’aléa dans le calcul.

↪ Modèles et analyses *mathématiques*.

EXEMPLE: le hachage

Stocker x à l'adresse $h(x)$.

Fichier de , , ,  ...





- Le choix d’une “bonne” fonction donne un *pseudo-aléa*.
- Probabilités classiques: *allocations aléatoires* n (objets) $\mapsto m$ (cases)

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

- Gestion de collisions: $\leadsto$ *combinatoire analytique*

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

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

2 ALGORITHMIQUE DES FLUX MASSIFS



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

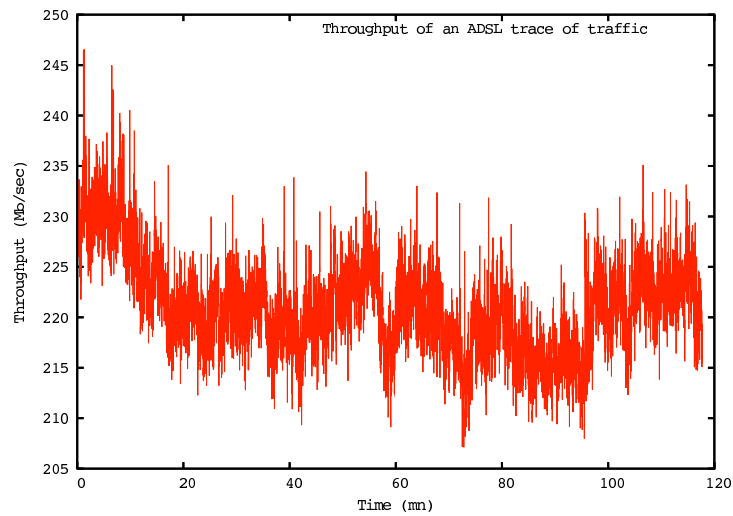


Google indexe 10 milliards de pages & prepare
100 Petabytes de données (10^{17} B).

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

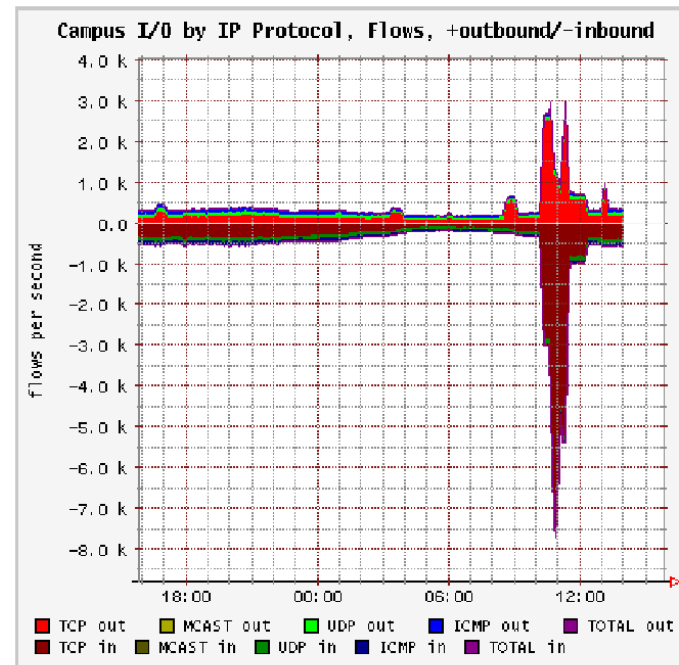
Cf. Leçon inaugurale.

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



(Trafic ADSL brut)

Volume brut



(Attaque)

Cardinalité

Exemple: Le problème de cardinalité

- Donnée: flux $s = s_1 s_2 \cdots s_\ell$, $s_j \in \mathcal{D}$, $\ell \propto 10^9$.
- Sortie: Estimation de la cardinalité n , $n \propto 10^7$.
- 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 > 10 ;
- **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	...	poitou	...
	3589			3589	

Le flux de données = un multi-ensemble $\leadsto$ réels uniformes $[0, 1]$

Une observable = une fonction de l'ensemble sous-jacent.

Une observable = une fonction de l'ensemble haché.

- A. On a vu passer le motif initial 0.011101
- B. Le minimum des valeurs lues est 0.00000001101001
- C. On a vu les motifs initiaux $0.1\dots$ et $0.01\dots$ et $0.001\dots$
- 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" 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.

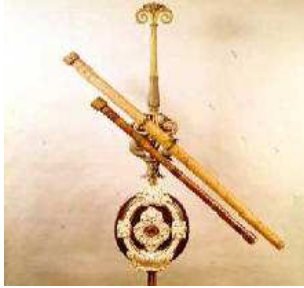
L'observable O vaut le maximum des positions du premier 1

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

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

≡ **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: $\mathbb{E}_n(O) = \xi + \text{oscillations} + o(1)$.

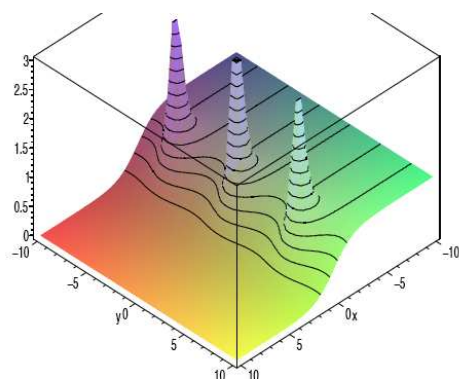
On *estime* la valeur logarithmique de n avec un *biais systématique* (φ) et une *dispersion* (ξ) de ± 1 ordre binaire de grandeur.

↪ corriger le biais; améliorer la précision!

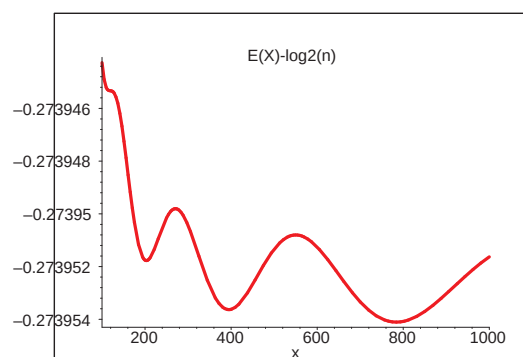


La transformation de Mellin: $\int_0^{\infty} f(x) x^{s-1} dx.$

- 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**

$h(x) = b_1 b_2 b_3 \dots$;

$\rho := \text{position}_{1\uparrow}(b_1 b_2 \dots)$;

$R := \max(R, \rho)$;

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 $\approx \pm \frac{1}{\sqrt{m}}$.

($m = 1000 \implies$ 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**
 $h(x) = b_1 b_2 \dots$; $A := \langle b_1 \dots b_{10} \rangle_{\text{base } 2}$;
 $\rho := \text{position}_{1\uparrow}(b_{11} b_{12} \dots)$;
 $R[A] := \max(R[A], \rho)$;
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 **bias non asymptotique**.

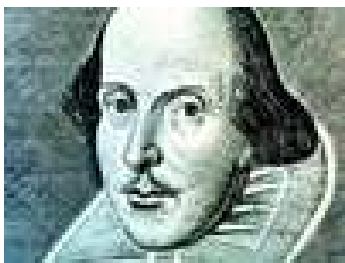
L'analyse mathématique (combinatoire, probabiliste, asymptotique) intervient de manière non triviale dans la conception.

(ici: Mellin + méthodes de col).

~ 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 ($m = 256$)



```
ghfffghfghgghgggghghheehfhhfhhgghghghhfgffffhhhiigfhhffgfiihfhhh  
igigighfgihffffghigihghigfhhgeegeghgghhhgghhfhidiigihighihehhfghg  
hfgighigffghdieghhhggghhfhghfiiheffghghihifgggffihgihfggighgiiif  
fjgfgjjhhjiihfjgehgghfhhfhjhiggghghihigghhihihgiighghflgjfgjjmfl
```

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 $R_1, \dots, R_m$ de $S \equiv$ **signature** de S .

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

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

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

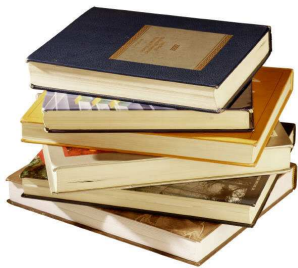
4.3 Comparaison de documents

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

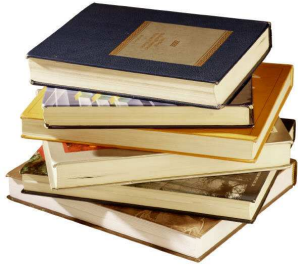
- la **taille** $\|S\|$ = nombre total d'éléments;
- la **cardinalité** $|S|$ = nombre d'éléments distincts.

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

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



*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?



$$\left\{ \begin{array}{l} |A| = \text{estim}(\text{sign}(A)) \\ |B| = \text{estim}(\text{sign}(B)) \\ |A \cup B| = \text{estim}(\text{sign}(A) \vee \text{sign}(B)) \end{array} \right. \quad \text{simil}(A, B) = \frac{|A| + |B| - |A \cup B|}{|A \cup B|}.$$

Soit une bibliothèque 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 $\simeq V + N^2$.

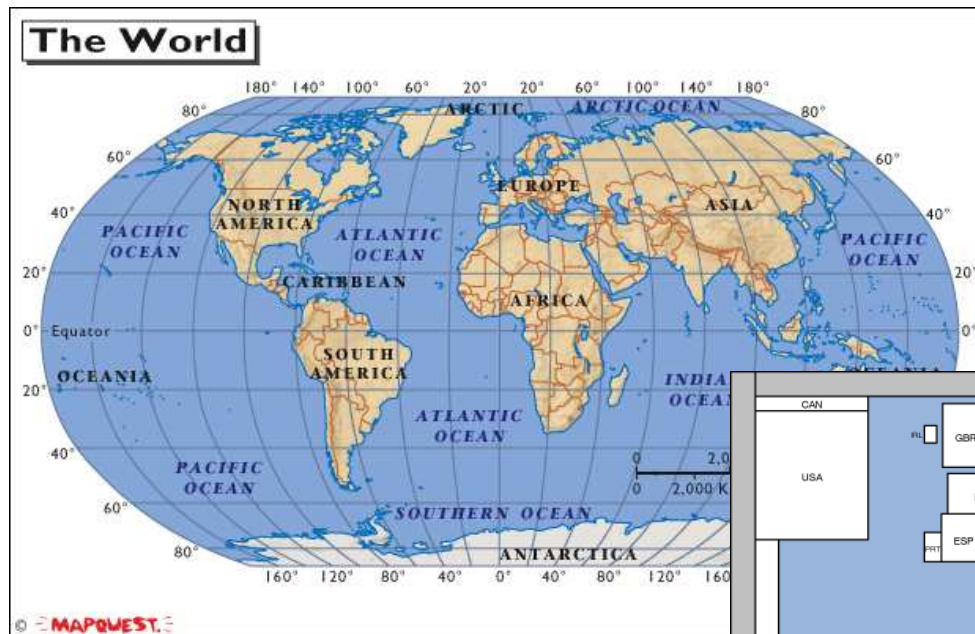
Match: signatures = 10^{12} contre exact = 10^{17} .

5 ÉCHANTILLONNAGE ADAPTATIF

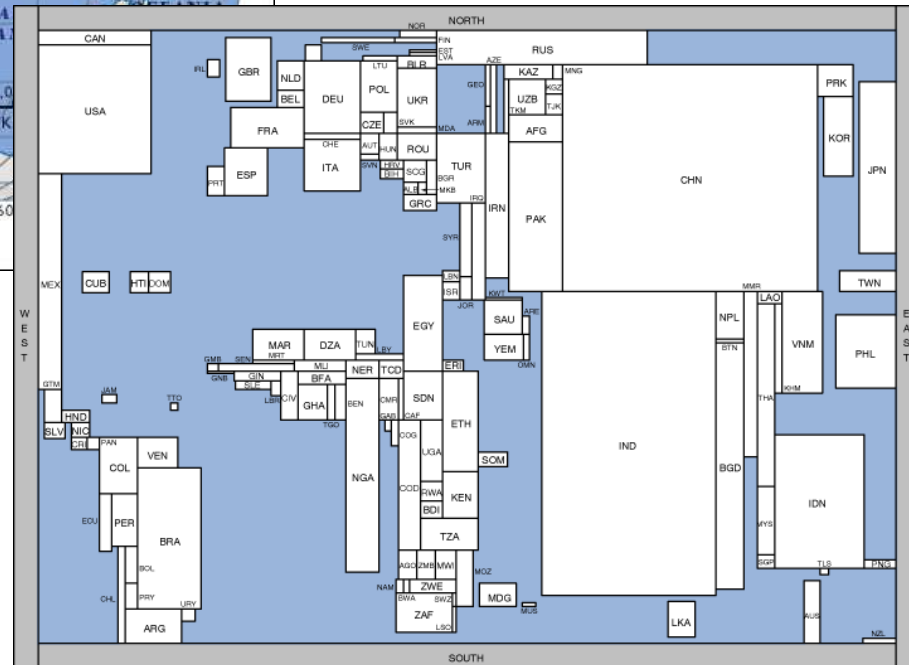


Peut-on localiser le centre de gravité géographique de la France; donnée: 60 millions de <personnes & communes>?

- **Exact**: oui bien sûr = élimination des doublons (“projection”)
- **Approché**: échantillonnage simple $\Rightarrow$ **au Sud Est de Paris(!!)**.

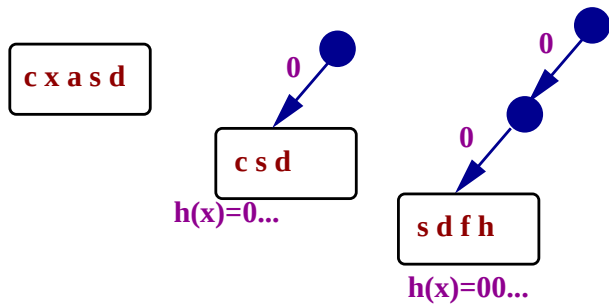


© Bettina Speckmann, TU Eindhoven)



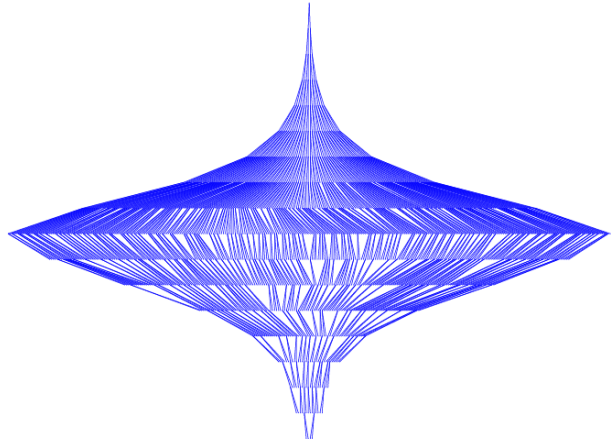
*Échantillonner sur le domaine des valeurs **distinctes**?*

L'échantillonnage adaptatif:



Algorithme: Adaptive Sampling(S : flux);
 $C := \emptyset$; {cache de capacité m }
 $p := 0$; {profondeur}
pour $x \in S$ **faire**
 si $h(x) = 0^p \dots$ **alors** $C := C \cup \{x\}$;
 si débordé(C) **alors** $p := p+1$; filtrer C ;
retourner C { $\approx m/2 \dots m$ 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**:

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



Hamlet

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

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

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

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

danskers, distract, fine, fra, immediately, loses, martiall, organe, pas-seth, pendant

Google = 8, tous vers Shakespeare/ Hamlet $\rightsquigarrow$ 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.

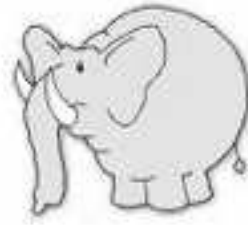
Cache de taille = 100, donne un échantillon de 79 éléments.

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

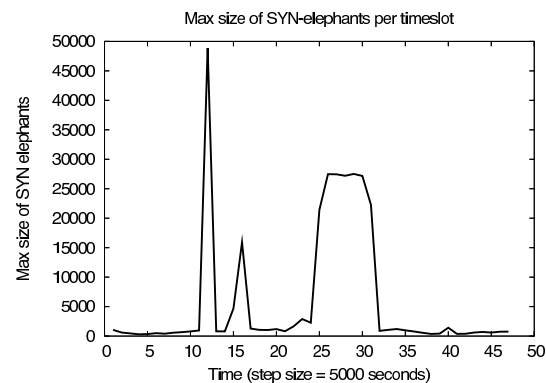
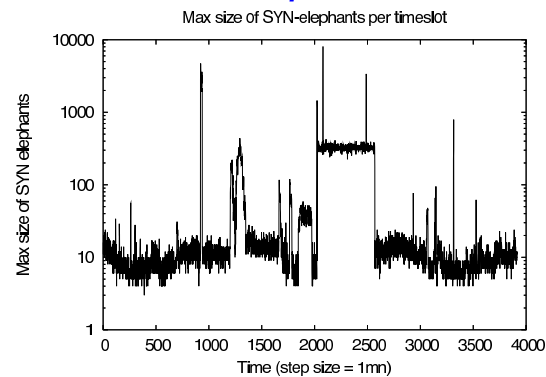
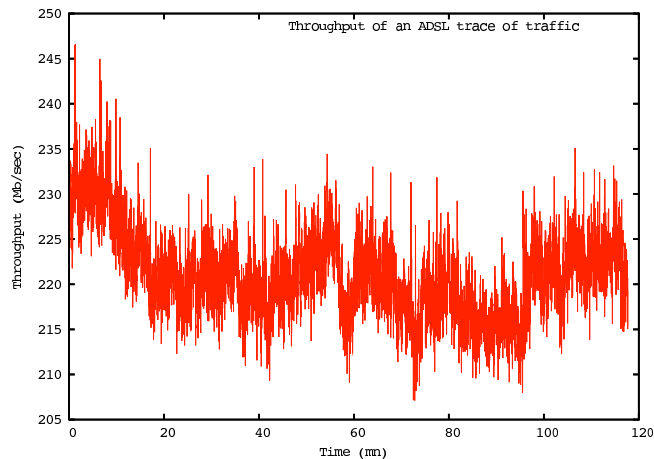
	1-Souris	2-Souris	3-Souris
<i>Estimé</i>	63%	17%	5%
<i>Réel</i>	60%	14%	6%

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 $\geq k$.



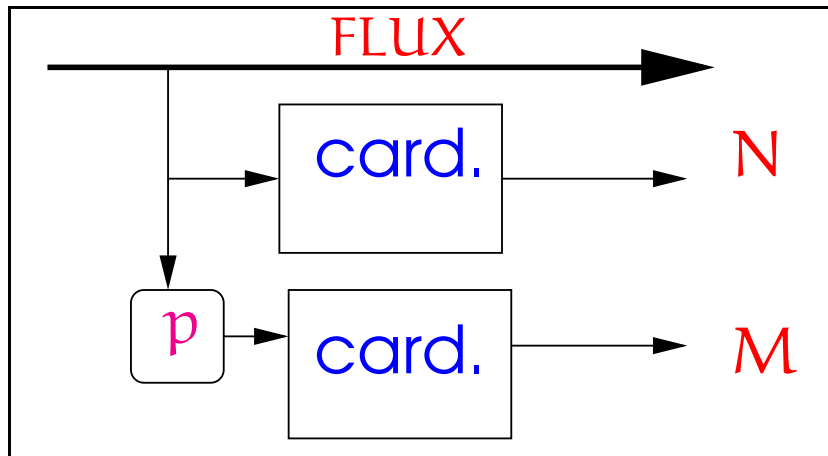
Attaques réseaux par déni de service (Y. Chabchoub, Ph. Robert)

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



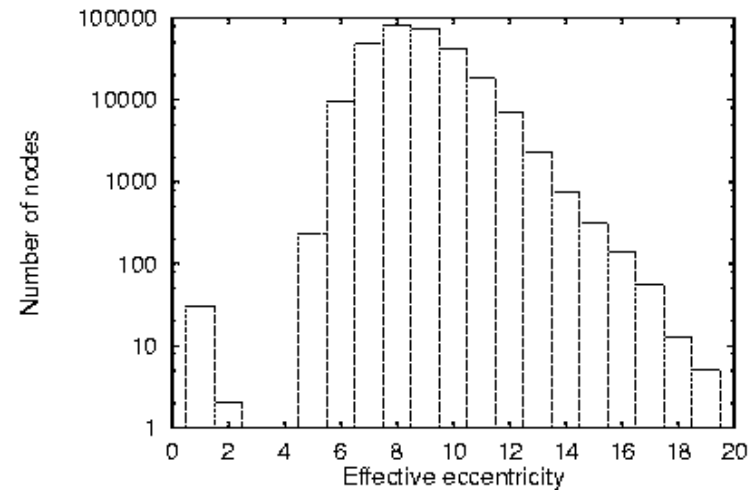
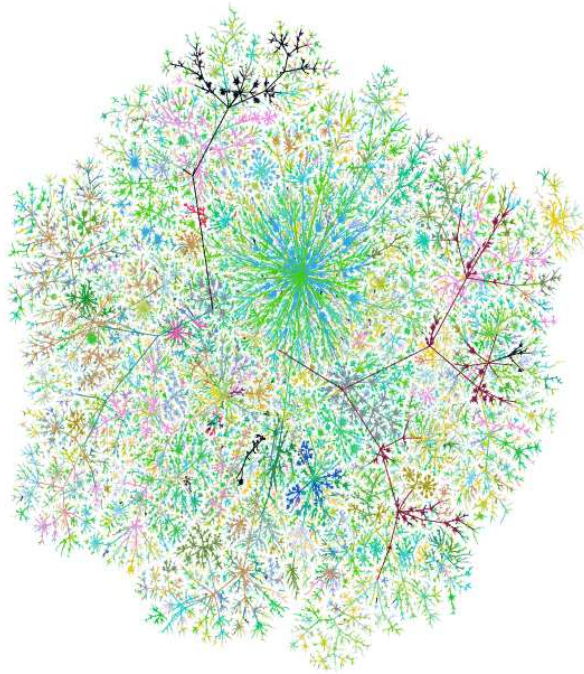
$$\begin{cases} N = N_s + N_e + \text{bruit} \\ M = \frac{1}{10}N_s + 0.65N_e + \text{bruit} \end{cases} \quad (p = \frac{1}{10})$$

Solution:

$$N_e \approx \frac{10M - N}{5.5}$$

(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

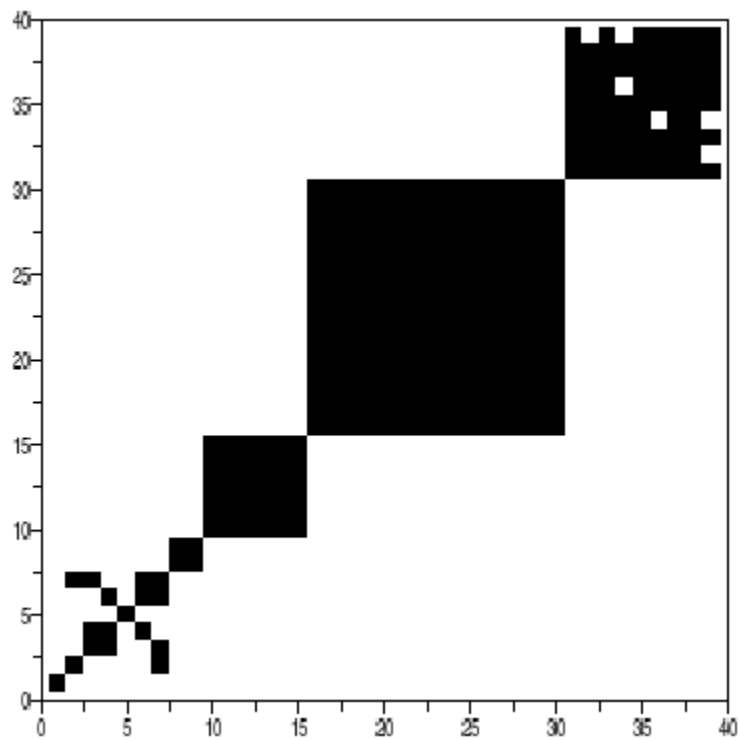
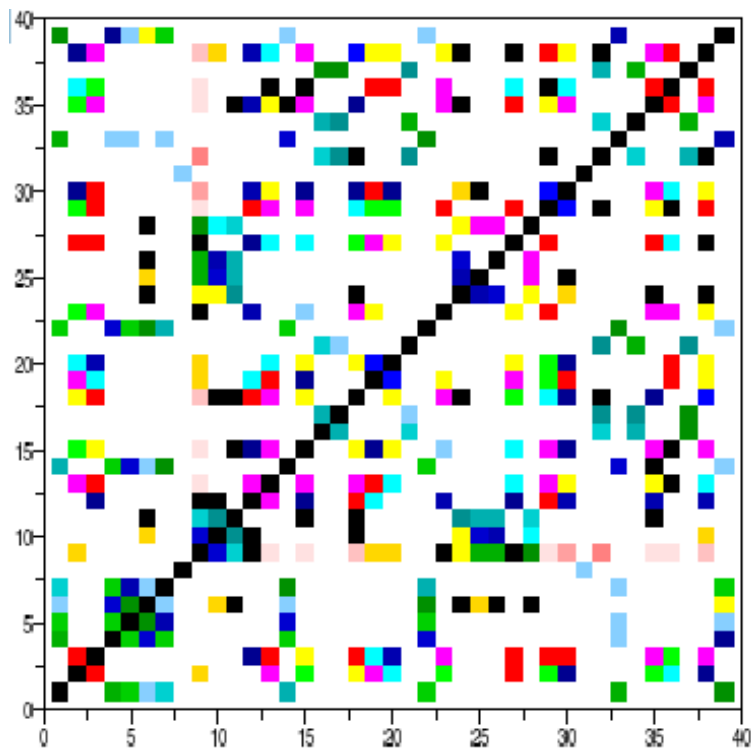


b) Histogram of diameters

Gain: $\times 300$.

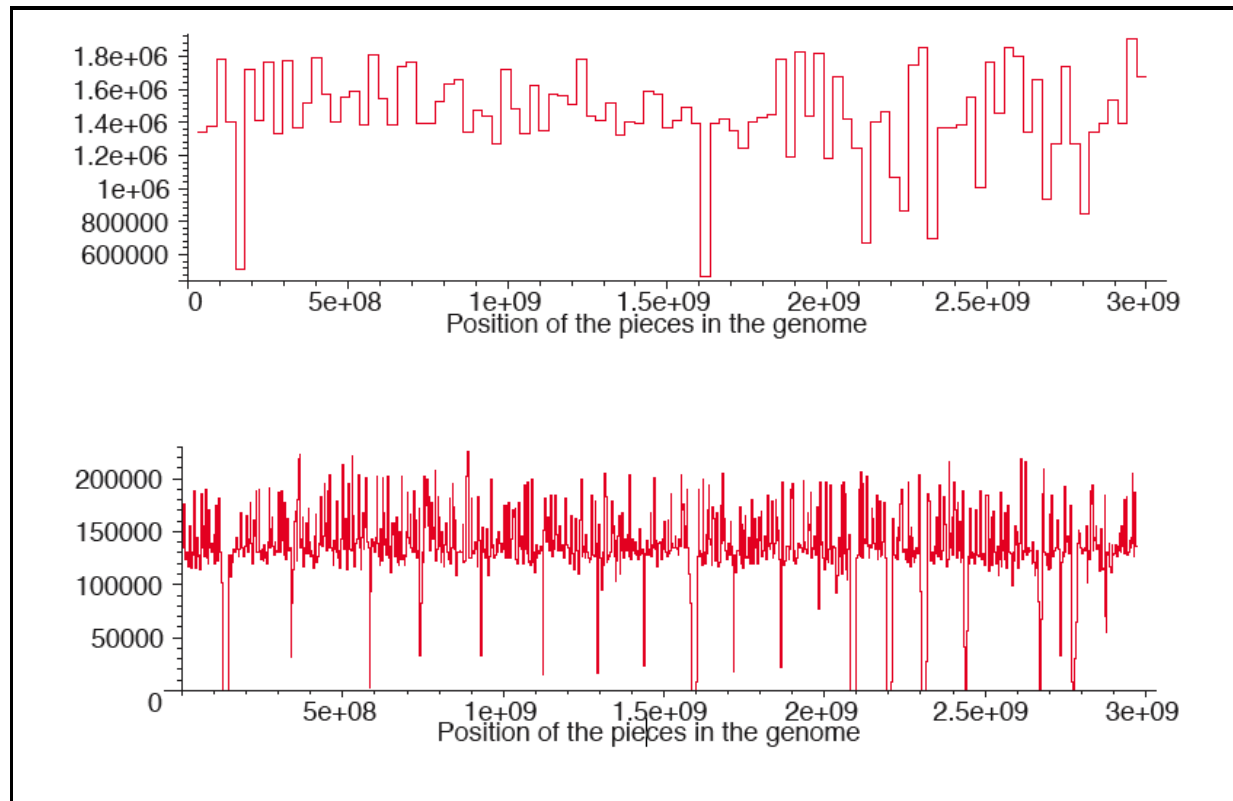
(Palmer, Gibbons, Faloutsos², Siganos 2001) Internet graph: 285k nodes, 430k edges.

Combien de langues?



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

Génome



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

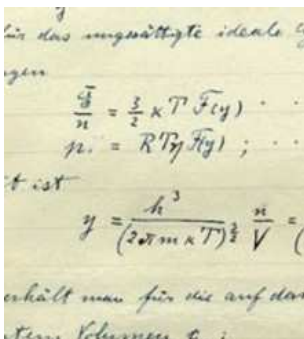
Conclusions (?)



Interprétation des résultats = un autre métier.



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



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