INFORMATION THEORY: MODELS, ALGORITHMS, ANALYSIS

REALISTIC ANALYSIS OF SORTING AND SEARCHING ALGORITHMS

Brigitte Vallée
GREYC Laboratory
(CNRS and University of Caen, France)

Mini-course based on joint works with
Julien Clément, Jim Fill and Philippe Flajolet

Journées du GT Alea 2010

Plan of the talk.

- Motivations : Realistic analyses of Sorting and Searching algorithms.
- A general model of source
- Description of the main results
- Exact analysis in the Poisson model : the `Trie`
- Exact analysis in the Poisson model : `QuickSort` and `QuickSelect` .
- Exact analyses in the Bernoulli model
- Asymptotic analysis : different types of sources.

Plan of the talk.

– Motivations : Realistic analyses of Sorting and Searching algorithms.

– A general model of source

– Description of the main results

– Exact analysis in the Poisson model : the `Trie`

– Exact analysis in the Poisson model : `QuickSort` and `QuickSelect`

– Exact analysis in the Bernoulli model

– Asymptotic analysis : different types of sources.

The classical framework for sorting and searching.

The main sorting algorithms or searching algorithms

e.g., `QuickSort`, `BST-Search`, `InsertionSort`,...

deal with n (distinct) keys $U_1, U_2, \ldots, U_n$ of the same ordered set Ω.

They perform comparisons and exchanges between keys.

The unit cost is the key–comparison.

The behaviour of the algorithm (wrt to key–comparisons)

only depends on the relative order between the keys.

It is sufficient to restrict to the case when $\Omega = [1..n]$.

The input set is then $\mathfrak{S}_n$, with uniform probability.

Then, the analysis of all these algorithms is very well known,

with respect to the number of key–comparisons performed

in the worst-case, or in the average case.

Here, realistic analysis of the two algorithms QuickSort and QuickSelect

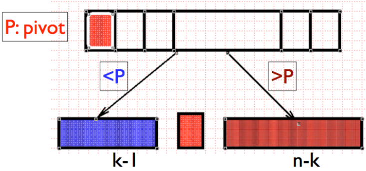
QuickSort (n, A): sorts the array A

Choose a pivot;

$(k, A_-, A_+) :=$ Partition(A);

QuickSort $(k-1, A_-)$;

QuickSort $(n-k, A_+)$.



QuickSelect (n, m, A): returns the value of the element of rank m in A.

Choose a pivot;

$(k, A_-, A_+) :=$ Partition(A);

If $m = k$ then QuickSelect := pivot

else if $m < k$ then QuickSelect $(k-1, m, A_-)$

else QuickSelect $(n-k, m-k, A_+)$;

Known results for QuickSort and QuickSelect for various values of rank m

about the mean number $K(n)$ of key–comparisons

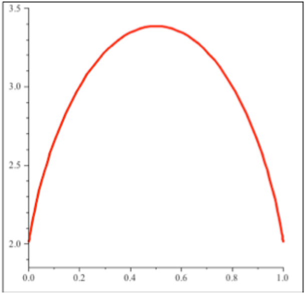
QuickSort (n)	sorts		$K(n) \sim 2n \log n$
QuickMin(n)	minimum	$m = 1$	$K(n) \sim 2n$
QuickMax(n)	maximum	$m = n$	$K(n) \sim 2n$
QuickRand(n)		$m \in [1..n]_{\mathcal{R}}$	$K(n) \sim 3n$
QuickQuant$_\alpha(n)$	α–quantile	$m = \lfloor \alpha n \rfloor$	$K(n) \sim \kappa(\alpha)\, n$
QuickMed(n)	median	$m = \lfloor n/2 \rfloor$	$K(n) \sim 2(1 + \log 2)n$

On the right,

the function $\kappa : \alpha \mapsto 2\,[1 + h(\alpha)]$

where $h(\cdot)$ is the entropy function

$h(\alpha) = \alpha |\log \alpha| + (1 - \alpha)|\log(1 - \alpha)|$



A more realistic framework for sorting.

Keys are viewed as words. The domain Ω of keys is a subset of Σ^∞,

$$\Sigma^\infty = \{\text{the infinite words on some ordered alphabet } \Sigma\}.$$

The words are compared [wrt the lexicographic order].

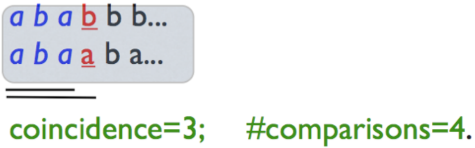
The realistic unit cost is now the symbol–comparison.

The realistic cost of the comparison between two words A and B,

$$A = a_1\, a_2\, a_3 \ldots a_i \ldots \qquad \text{and} \qquad B = b_1\, b_2\, b_3 \ldots b_i \ldots$$

equals $k+1$, where k is the length of their largest common prefix

$$k := \max\{i; \quad \forall j \le i, \quad a_j = b_j\} = \text{the coincidence}$$



We are interested in this new cost for each algorithm:

the number of symbol–comparisons ... and its mean value $S(n)$ (for n words)

How is $S(n)$ compared to $K(n)$? That is the question....

An initial question asked by Sedgewick in 2000...

... In order to also compare with other text algorithms.

Two data structures for sorting a set of words

— the trie, for dictionary algorithms

— the binary search tree (BST) closely related to `QuickSort`

The $\mathtt{Trie}$ structure

A finite set $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$ formed with n words.

The tree $\mathtt{Trie}\,(\mathcal{X})$ built on $\mathcal{X}$ is defined by the three rules:

– If $|\mathcal{X}| = 0$, $\mathtt{Trie}\,(\mathcal{X}) = \emptyset$

– If $|\mathcal{X}| = 1$, $\mathcal{X} = \{X\}$, $\mathtt{Trie}\,(\mathcal{X})$ is a leaf labeled by X.

– If $|\mathcal{X}| \geq 2$, then $\mathtt{Trie}\,(\mathcal{X})$ is formed with

  – an internal node

  – and n subtries $\mathtt{Trie}\,(\mathcal{X} \setminus m_1), \ldots, \mathtt{Trie}\,(\mathcal{X} \setminus m_r)$

  where $\mathcal{X} \setminus m :=$ {words of $\mathcal{X}$ that begin with m, stripped of m}.

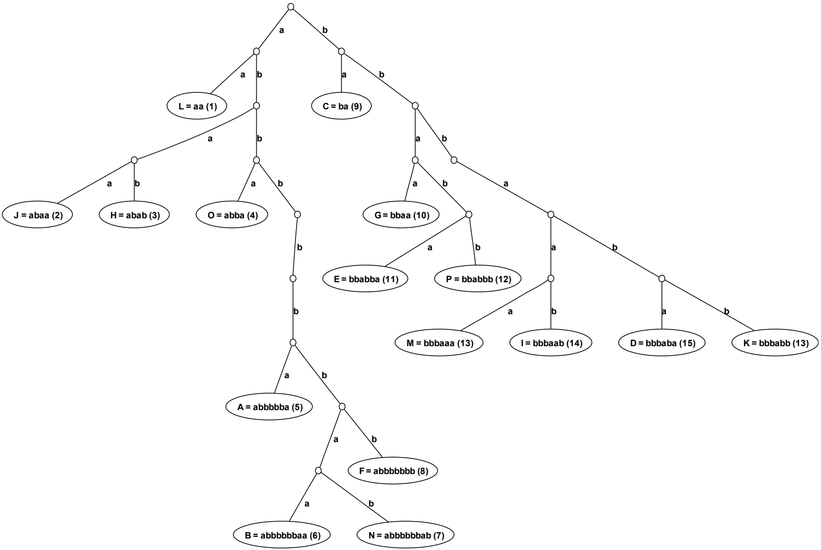
If $\mathcal{X} \setminus m \neq \emptyset$, the edge: internal node $\rightarrow \mathtt{Trie}\,(\mathcal{X} \setminus m)$ has label m.

The trie structure – An example : A trie built on a set of words.

A = abbbbbaaabab B = abbbbbbaabaa C = baabbbabbbba D =bbbababbbaab E = bbabbaababbb

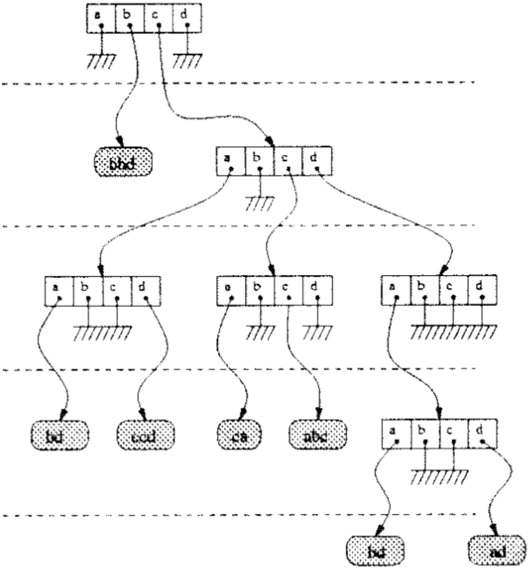
F = abbbbbbbbabb G = bbaabbabbaba H = ababbbabbbab I = bbbaabbbbbbb J = abaabbbbaabb

K = bbbabbbbbbaa L = aaaabbabaaba M = bbbaaabbbbbb N = abbbbbbabbaa O = abbababaabbbb P = bbabbbaaaabb



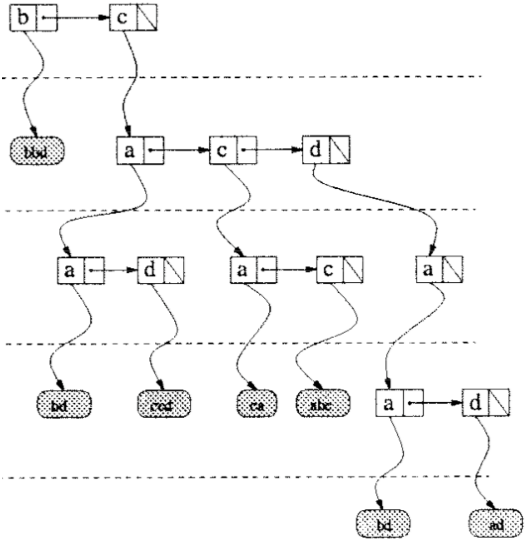
Study of Tries – Various implementations – The array–trie

Size? Path-length of the array-trie?



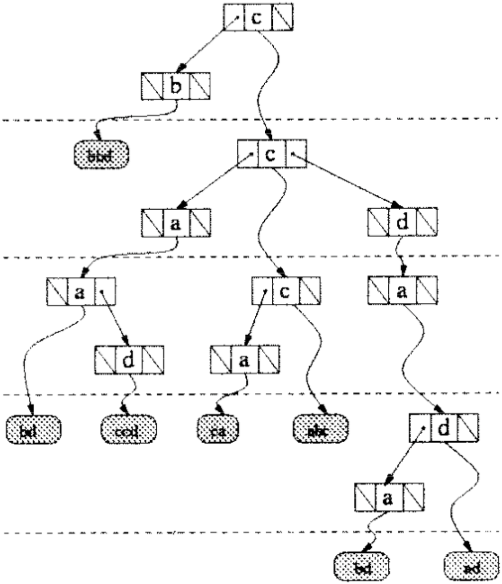
Study of Tries – Various implementations –The list–trie

Path-length of the list-trie?



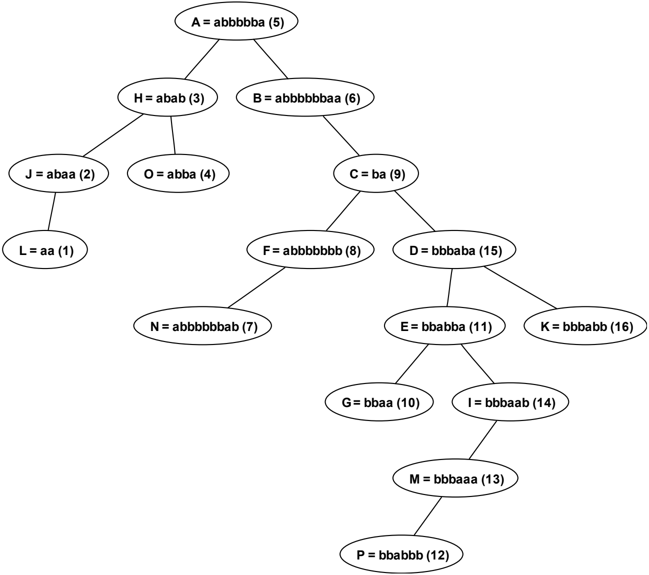
Study of Tries – Various implementations –The bst–trie or the ternary search trie

Path-length of the bst-trie?



The BST (binary search tree) built on the same sequence of words

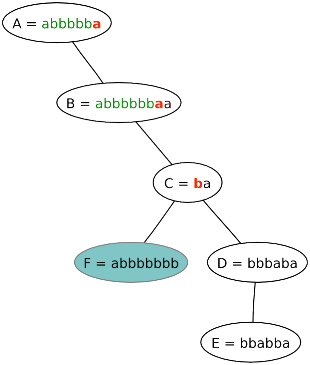
A = abbbbbaaabab B = abbbbbbaabaa C = baabbbabbbba D = bbbababbbaab E = bbabbaababbb
F = abbbbbbbbabb G = bbaabbababa H = ababbbabbbab I = bbbaabbbbbbb J = abaabbbbaabb
K = bbbabbbbbbaa L = aaaabbabaaba M = bbbaaabbbbbb N = abbbbbbabbaa O = abbababab bbb P = bbabbbaaaabb



What is the symbol-path-length of a BST ?

An example : The cost of the insertion of the key F into the BST

$F =$abbbbbbb



Number of symbol comparisons needed $= 16$

$= 7$ for comparing to A

$+\ 8$ for comparing to B

$+\ 1$ for comparing to C

Plan of the talk.

– Motivations : Realistic analyses of Sorting and Searching algorithms.

– A general model of source

– Description of the main results

– Exact analysis in the Poisson model : the `Trie`

– Exact analysis in the Poisson model : `QuickSort` and `QuickSelect` .

– Exact analysis in the Bernoulli model

– Asymptotic analysis : different types of sources.

The parametrization of a general source

A general source $\mathcal{S}$ produces infinite words

on an ordered alphabet $\Sigma := \{a_1, \ldots, a_r\}$.

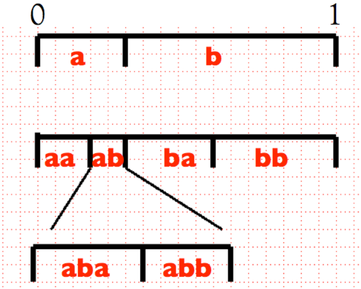
For $w \in \Sigma^\star$, $p_w :=$ probability that a word begins with the prefix w.

The set $\{p_w, \quad w \in \Sigma^\star\}$ defines the source $\mathcal{S}$. We assume

$$\pi_k := \sup\{p_w, \quad w \in \Sigma^k\} \to 0 \quad \text{for } k \to \infty$$

For each length k, we consider the p_w's for $w \in \Sigma^k$,

sorted with respect to the lexicographic order on Σ^k.



We define two other probabilities

$$p_w^{(-)} := \sum_{\substack{\alpha \in \Sigma^k, \\ \alpha < w}} p_\alpha, \quad p_w^{(+)} := \sum_{\substack{\alpha \in \Sigma^k, \\ \alpha > w}} p_\alpha.$$

Then, for any $X \in \Sigma^\infty$,

$$\lim_{w \to X} p_w^{(-)} = 1 - \lim_{w \to X} p_w^{(+)} := P(X)$$

Consider the set $\mathcal{L}(\mathcal{S}) \subset \Sigma^\infty$ the set of infinite words emitted by $\mathcal{S}$.

The function $P : \mathcal{L}(\mathcal{S}) \to [0,1]$ is strictly increasing .....

outside a denumerable exceptional set

$$\mathcal{E} := \{X \in \mathcal{L}(\mathcal{S}); \quad \exists Y \in \mathcal{L}(\mathcal{S}) \quad \text{with} \quad Y \neq X, \quad P(X) = P(Y)\}$$

Outside this exceptional set, each infinite word X is written as

$$X = M(u) \text{ with } M : [0,1] \to \mathcal{L}(\mathcal{S}).$$

The map M provides a parametrization of the source $\mathcal{S}$.

Via the mapping M,

[Drawing in $\mathcal{S}$ wrt the p_w's] $\equiv$ [Uniform drawing in $[0,1]$]

For any finite prefix $w \in \Sigma^\star$,

the set $\{u, \ M(u) \text{ begins with } w\}$ is an interval with endpoints $p_w^{(-)}, p_w^{(+)}$.

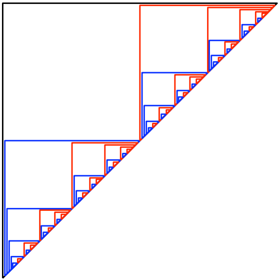
This is the fundamental interval of w. Its length equals p_w.

For any finite prefix $w \in \Sigma^\star$,

the set $\{u,\ M(u) \text{ begins with } w\}$ is an interval with endpoints $p_w^{(-)}, p_w^{(+)}$.

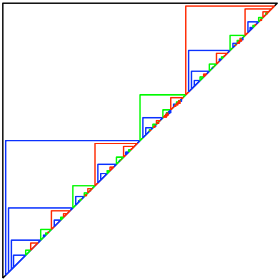
This is the fundamental interval of w. Its length equals p_w.

Instances of fundamental intervals for two memoryless sources.



Memoryless source on $\{a, b\}$

$p_a = 1/2$, $p_b = 1/2$

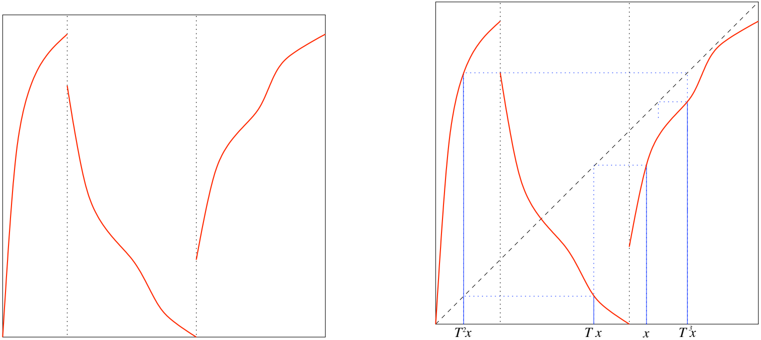


Memoryless source on $\{a, b, c\}$

$p_a = 1/2$, $p_b = 1/6$, $p_c = 1/3$

Natural instances of sources: Dynamical sources

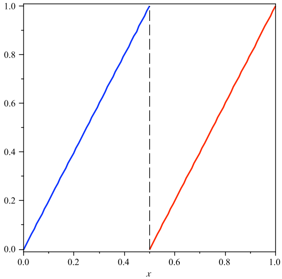
With a shift map $T : \mathcal{I} \to \mathcal{I}$ and an encoding map $\tau : \mathcal{I} \to \Sigma$,
the emitted word is $M(x) = (\tau x, \tau T x, \tau T^2 x, \dots \tau T^k x, \dots)$

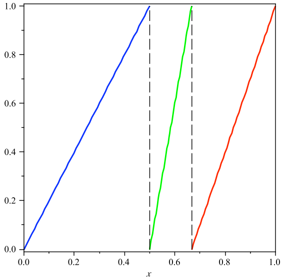


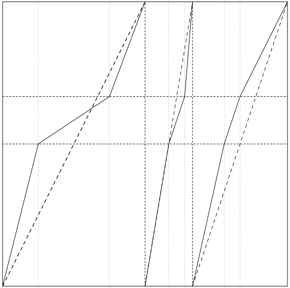
A dynamical system, with $\Sigma = \{a, b, c\}$ and a word $M(x) = (c, b, a, c \dots)$.

Memoryless sources or Markov chains.

= Dynamical sources with affine branches....



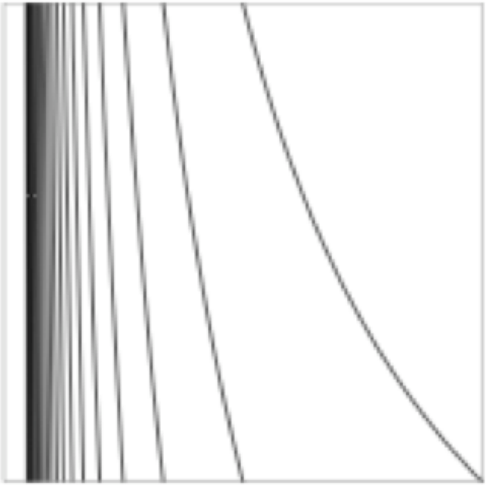




The dynamical framework leads to more general sources.

The position and the curvature of branches entail correlation between symbols

Example : the Continued Fraction source



A main analytical object related to any source:

the Dirichlet series of probabilities, $\Lambda(s) := \sum_{w \in \Sigma^\star} p_w^s$

Memoryless sources, with probabilities (p_i)

$$\Lambda(s) = \frac{1}{1-\lambda(s)} \qquad \text{with} \quad \lambda(s) = \sum_{i=1}^{r} p_i^s$$

Markov chains, defined by – the vector R of initial probabilities (r_i)

– and the transition matrix $P := (p_{i,j})$

$$\Lambda(s) = {}^t\mathbf{1}(I - P(s))^{-1}R(s) \qquad \text{with} \quad P(s) = (p_{i,j}^s), \quad R(s) = (r_i^s).$$

A general dynamical source

$\Lambda(s)$ closely related to $(I - \mathbf{H}_s)^{-1}$

where $\mathbf{H}_s$ is the (secant) transfer operator of the dynamical system.

Plan of the talk.

– Motivations : Realistic analyses of Sorting and Searching algorithms.

– A general model of source

– Description of the main results.

– Exact analysis in the Poisson model : the `Trie`

– Exact analysis in the Poisson model : `QuickSort` and `QuickSelect` .

– Exact analysis in the Bernoulli model

– Asymptotic analysis : different types of sources.

What is already known about the mean number of symbol-comparisons?

The `Trie` structure is very well-studied, but only for particular sources:

the so–called simple sources: memoryless or Markov chains

and only for the array–trie.

The number of symbols comparaisons used in `QuickSort`, and `QuickSelect`, is already studied by Janson, Fill, Nakama ('06), but only

– in the case of memoryless sources,

– for `QuickSort, QuickMin, QuickMax, QuickRand`

Here, we study the mean number of symbol-comparisons,

for a general source and a general algorithm/structure of the class.

– There are precise restrictive hypotheses on the source,
and sufficient conditions under which these hypotheses hold.

– We provide a closed form for the constants of the analysis,
for any source of the previous type.

– We use different methods, with limited computation...

Case of $\mathtt{Trie}(n)$ [CFV 01]

Theorem 1. *For any Λ–tame source,*
the mean size $R(n)$ and the mean path length $C(n)$ of an array–trie built on n words independently drawn from the source satisfy

$$R(n) \sim \frac{1}{h_{\mathcal{S}}}\, n \qquad C(n) \sim \frac{1}{h_{\mathcal{S}}}\, n \log n.$$

and involve the entropy $h_{\mathcal{S}}$ of the source $\mathcal{S}$, defined as

$$h_{\mathcal{S}} := \lim_{k\to\infty} \left[\frac{-1}{k} \sum_{w \in \Sigma^k} p_w \log p_w \right],$$

where p_w is the probability that a word begins with prefix w.

Case of $\texttt{Trie}(n)$ [CFV 01]

Theorem 2. *For any Λ–tame stationary source,*
– the mean path-length $L(n)$ of a list–trie
– the mean path length $A(n)$ of an bst–trie
built on n words independently drawn from the source satisfy

$$L(n) \sim \frac{K_L(\mathcal{S})}{h_{\mathcal{S}}}\, n \qquad A(n) \sim \frac{K_A(\mathcal{S})}{h_{\mathcal{S}}}\, n \log n.$$

and involve the entropy $h_{\mathcal{S}}$, together with constants

$$K_L(\mathcal{S}) = \sum_{i \in \Sigma} P_{[>i]} \qquad K_A(\mathcal{S}) = 2 \sum_{\substack{i,j \in \Sigma \\ i<j}} \frac{p_i p_j}{P_{[i,j]}}$$

where p_i is the probability that a word begins with symbol i
and $P_{[i,j]} := \sum_{k=i}^{j} p_k$.

Case of `QuickSort`(n) or `BST`(n) [CFFV 08]

Theorem 3. *For any Λ–tame source,*
the mean number $B(n)$ of symbol comparisons used by `QuickSort`(n)
(or the mean number of symbols comparisons used to built the BST)
on n words of the source satisfies

$$B(n) \sim \frac{1}{h_{\mathcal{S}}}\, n\, \log^2 n.$$

and involves the entropy $h_{\mathcal{S}}$ of the source $\mathcal{S}$, defined as

$$h_{\mathcal{S}} := \lim_{k\to\infty} \left[\frac{-1}{k} \sum_{w\in\Sigma^k} p_w \log p_w \right],$$

where p_w is the probability that a word begins with prefix w.

Compared to $K(n) \sim 2n \log n$, there is an extra factor equal to $1/(2h_{\mathcal{S}}) \log n$

Compared to $C(n) \sim (1/h_{\mathcal{S}})\, n\, \log n$, there is an extra factor of $\log n$.

Case of $\texttt{QuickQuant}_\alpha(n)$ [CFFV 09]

Theorem 4. *For any* Π*–tame source,*
the mean number of symbol comparisons used by $\texttt{QuickQuant}_\alpha(n)$
satisfies

$$Q(n)^{(\alpha)} \sim \rho_{\mathcal{S}}(\alpha)\, n \qquad \rho_{\mathcal{S}}(\alpha) = \sum_{w\in\Sigma^\star} p_w\, L\left(\frac{|\alpha - \mu_w|}{p_w}\right).$$

$$\mu_w = \frac{1}{2}\left[p_w^{(+)} + p_w^{(-)}\right] = \text{the middle of the fundamental interval}$$

The function L is an even function given by $L(y) = 2[1 + H(y)]$,

$$H(y) = \begin{cases} -(y^+ \log y^+ + \ y^- \log y^-), & \text{if } 0 \le y < 1/2 \\ 0, & \text{if } y = 1/2 \\ y^+(\log|y^+| - \log|y^-|), & \text{if } y > 1/2. \end{cases}$$

$H(y)$ is a modified entropy function expressed with $y^+ := (1/2) + y$, $y^- = (1/2) - y$.

Some particular cases for the constant $\rho_{\mathcal{S}}(\alpha)$.

Constants for `QuickMin` $(\alpha = 0 \to \epsilon = +)$ and `QuickMax` $(\alpha = 1 \to \epsilon = -)$

$$c_{\mathcal{S}}^{(\epsilon)} := 2 \sum_{w \in \Sigma^\star} p_w \left[1 - \frac{p_w^{(\epsilon)}}{p_w} \log \left(1 + \frac{p_w}{p_w^{(\epsilon)}} \right) \right].$$

Constant for `QuickRand` $\underline{c}_{\mathcal{S}} = \int_0^1 \rho_{\mathcal{S}}(\alpha) d\alpha$

$$\underline{c}_{\mathcal{S}} = \sum_{w \in \Sigma^\star} p_w^2 \left[2 + \frac{1}{p_w} + \sum_{\epsilon = \pm} \left[\log \left(1 + \frac{p_w^{(\epsilon)}}{p_w} \right) - \left(\frac{p_w^{(\epsilon)}}{p_w} \right)^2 \log \left(1 + \frac{p_w}{p_w^{(\epsilon)}} \right) \right] \right]$$

The constants of the analysis for the binary source.

$$h_{\mathcal{B}} = \log 2, \qquad c_{\mathcal{B}}^{(+)} = c_{\mathcal{B}}^{(-)} = c_{\mathcal{B}}^{(\epsilon)}$$

$$c_{\mathcal{B}}^{(\epsilon)} = 4 + 2\sum_{\ell \geq 0} \frac{1}{2^{\ell}} + 2\sum_{\ell \geq 0} \frac{1}{2^{\ell}} \sum_{k=1}^{2^{\ell}-1} \left[1 - k\log\left(1 + \frac{1}{k}\right)\right]$$

$$\underline{c}_{\mathcal{B}} = \frac{14}{3} + 2\sum_{\ell=0}^{\infty} \frac{1}{2^{2\ell}} \sum_{k=1}^{2^{\ell}-1} \left[k + 1 + \log(k+1) - k^2\log\left(1 + \frac{1}{k}\right)\right]$$

Numerically, $\quad c_{\mathcal{B}}^{(\epsilon)} = 5.27937......, \qquad c_{\mathcal{B}} = 8.20731......$

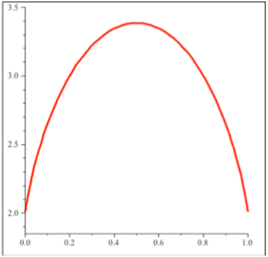
To be compared to the constants of the number of key–comparisons

$\kappa = 2 \qquad$ or $\qquad \kappa = 3$

The curve $\alpha \mapsto \rho(\alpha)$ is a fractal deformation of $\alpha \mapsto \kappa(\alpha)$

$\kappa(\alpha)$ the constant of the number of key–comparisons in $\texttt{QuickQuant}_\alpha$

The plot of $\alpha \mapsto \kappa(\alpha)$



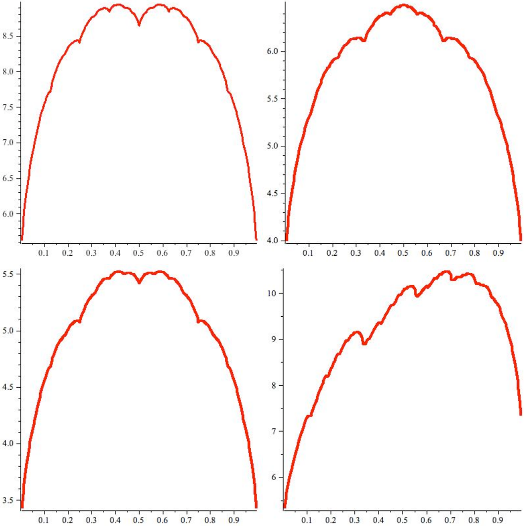
..... To be compared

to the plots of $\alpha \mapsto \rho(\alpha)$

for four memoryless sources

– three unbiased, $r = 2, 3, 4$

– one biased (1/3, 2/3)



What about the function $\alpha \mapsto \rho_{\mathcal{S}}(\alpha)$?

In the case where $\mathcal{S} =$ the unbiased memoryless source with r symbols.

$\rho_{\mathcal{S}}$ is denoted by ρ_r.

If r is odd, ρ_r is maximum at $\alpha = 1/2$ (case of `QuickMed`)

If r is even, this is not true. For which value of α, $\rho_r(\alpha)$ is maximum?

Is ρ_r differentiable? Is it Hölder?

When $r \to \infty$, $\rho_r(\alpha) \to 2[1 + h(\alpha)]$

$=$ the constant which intervenes in the mean number of key–comparisons.

($h(.)$ is the entropy function)

Plan of the talk.

- Motivations : Realistic analyses of Sorting and Searching algorithms.
- A general model of source
- Description of the main results.
- Exact analysis in the Poisson model : the `Trie`
- Exact analysis in the Poisson model : `QuickSort` and `QuickSelect` .
- Exact analysis in the Bernoulli model
- Asymptotic analysis : different types of sources.

Three main steps for the analysis
of the mean number $S(n)$ of symbol comparisons

(1) First step (algebraic).

The Poisson model $\mathcal{P}_Z$ deals with a variable number N of keys:
N is a random variable which follows a Poisson law of parameter Z.

We first obtain nice expressions for the mean number $\widetilde{S}(Z)$ ....

(2) Second step (algebraic).

It is then possible to return to the model where the number of keys is fixed.
We obtain a nice exact formula for $S(n)$ ....

from which it is not easy to obtain the asymptotics...

(3) Third step (analytic).

Then, the Rice formula provides the asymptotics of $S(n)$ ($n \to \infty$),

as soon as the source is "tame"

Λ–tame for `QuickSort` and `Tries` , Π–tame for `QuickSelect`

The three steps of the analysis

$$
\begin{tikzcd}
\text{Mean Value in the Poisson model} \arrow[r, Rightarrow, "\text{AlgDePo}"] \arrow[d, Rightarrow, "\text{Mellin}"'] & \text{Mean Value in the Bernoulli model} \arrow[d, Rightarrow, "\text{Rice}"] \\
\text{Asymptotic Mean Value in the Poisson model} \arrow[r, Rightarrow, "\text{AnDePo}"] & \text{Asymptotic Mean Value in the Bernoulli model}
\end{tikzcd}
$$

Two possible ways from the exact mean value in the Poisson model.
to the asymptotic mean value in the Bernoulli model.

Dealing with the Poisson Model $\mathcal{P}_z$

– The number N of keys is drawn according to the Poisson law

$$\Pr[N = n] = e^{-z}\frac{z^n}{n!},$$

– Then, the N words are independently drawn from the source.

or : N reals are uniformly and independently chosen in the unit interval

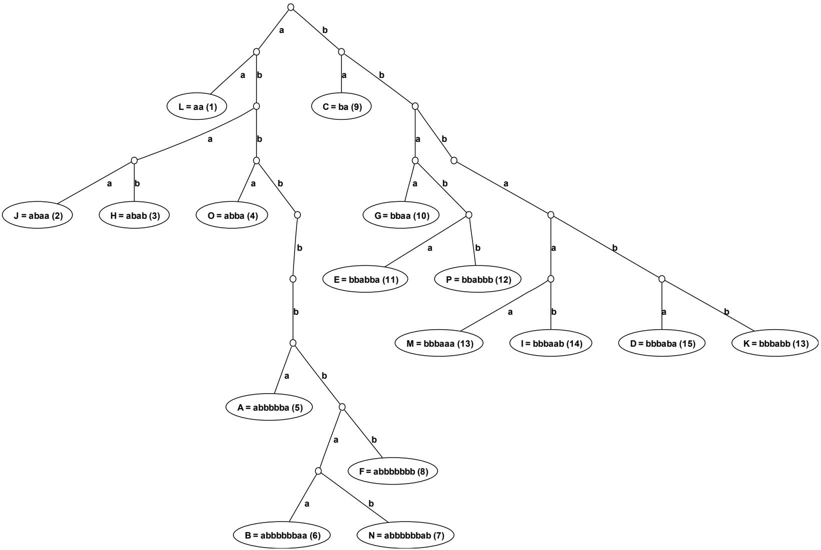
Two nice properties of the Poisson model.

about the number $N_{[a,b]}$ of words $M(v)$ with $v \in [a, b]$

(i) $N_{[a,b]}$ follows a Poisson law of parameter $z(b - a)$.

(ii) For $[a, b] \cap [c, d] = \emptyset$, the variables $N_{[a,b]}$ and $N_{[c,d]}$ are independent.

Study of Tries in the Poisson model



Study of Tries in the Poisson model

Main parameter on a node n_w labelled with prefix w:

$N_w :=$ the number of keys which begin with prefix w.

$N_w :=$ the number of keys which go through the node n_w

The size, and the path length of a plain trie (array-trie) equal

$$R = \sum_{w \in \Sigma^\star} \mathbf{1}_{[N_w \geq 2]} \qquad C = \sum_{w \in \Sigma^\star} \mathbf{1}_{[N_w \geq 2]} \cdot N_w,$$

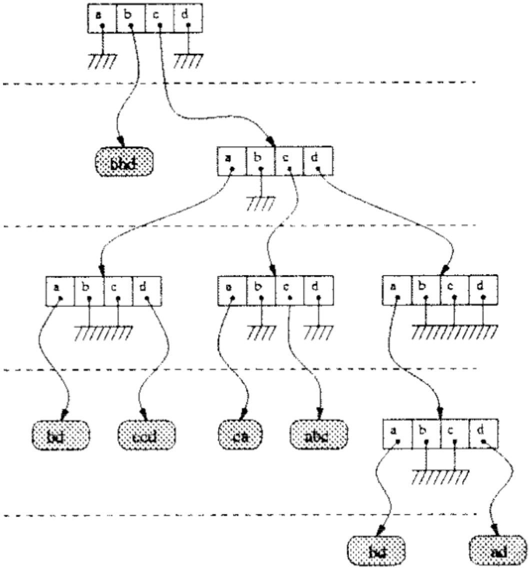
In the $\mathcal{P}_z$ model, the cardinality N_w follow a Poisson law of parameter zp_w

The mean size and the mean path-length are

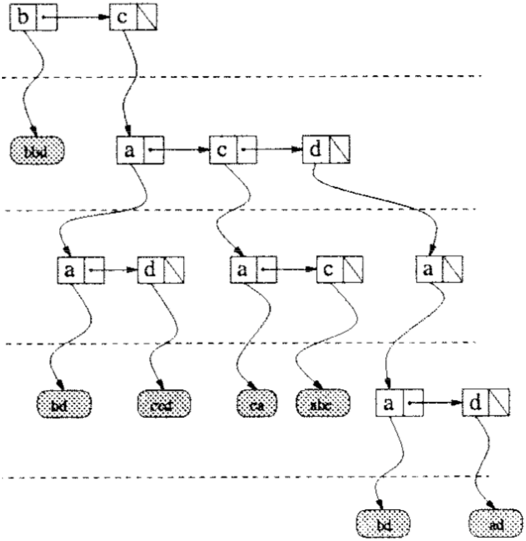
$$\widetilde{R}(z) = \sum_{w \in \Sigma^\star} 1 - (1 + zp_w)e^{-zp_w} \qquad \widetilde{C}(z) = \sum_{w \in \Sigma^\star} zp_w[1 - e^{-zp_w}].$$

Study of Tries in the Poisson model. Other implementations

The array–trie

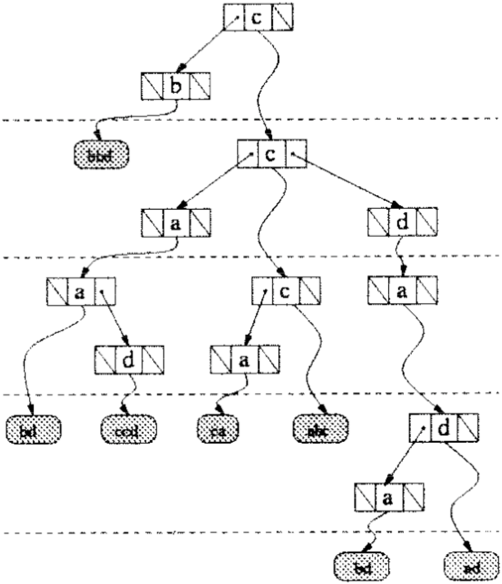


The list–trie

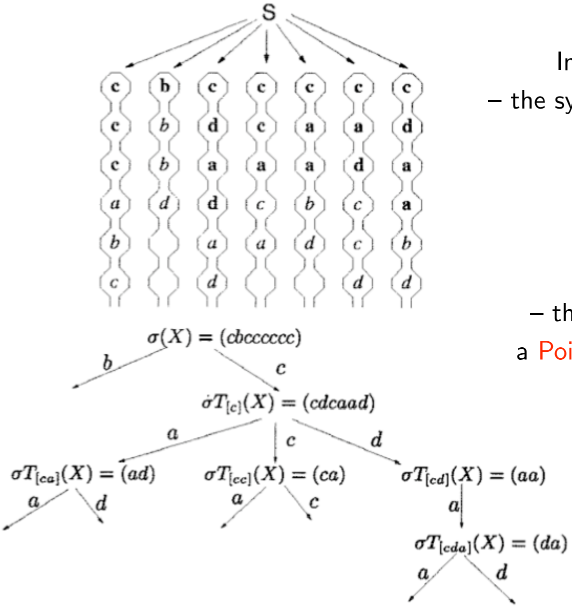


Study of Tries in the Poisson model. Other implementations

The bst–trie or the ternary search trie



Vertical (infinite) words versus horizontal finite slices.



In a node n_w with label w,

– the symbols of the slice are produced by a memoryless source with probabilities

$$p_{m|w} = \frac{p_{w.m}}{p_w}$$

– the number of symbols follows a Poisson law with parameter zp_w

The path-length inside a node depends on the data structure in the slice.

$N_i :=$ the number of symbols of the slice equal to a_i.

$N_{[i,j]} :=$ the number of symbols z with $\texttt{val}(z) \in [a_i, a_j]$

For the list-trie (L), or the bst-trie (A), the path-length in a slice is

$$\delta_L = \sum_{i \in \Sigma} N_i \sum_{j<i} \mathbf{1}_{[N_j \geq 1]} \qquad \delta_A = \sum_{i \in \Sigma} N_i \sum_{j \neq i} \mathbf{1}_{[a_j \text{ ancestor of } a_i \text{ in bst}]}$$

a_j is ancestor of a_i (with $j < i$) iff

$$\exists x \in \Sigma, \quad \texttt{val}(x) = a_j, \quad \texttt{ord}(x) = \min\{\texttt{ord}(z), \quad \texttt{val}(z) \in [a_i, a_j]\}$$

$$\Longrightarrow \qquad \Pr[\, a_j \text{ ancestor of } a_i \text{ in bst}] = \frac{2N_j}{N_{[i,j]}}$$

$$\Longrightarrow \qquad \underline{\delta}_A = 2 \sum_{\substack{(i,j)\in\Sigma^2 \\ i<j}} \frac{N_i N_j}{N_{[i,j]}} = 2 \sum_{\substack{(i,j)\in\Sigma^2 \\ i<j}} \frac{N_i N_j}{N_i + N_j + N_{]i,j[}}$$

Mean values of parameters in a slice

Assume that :

– the number of symbols follow a Poisson law $\mathcal{P}_z$

– the symbols are independently emitted with $p_i := \Pr[a_i]$

$$\mathbb{E}[\delta_L, \mathcal{P}_z, \mathcal{B}] = \sum_{j \in \Sigma} z\, P_{[>j]}\, (1 - e^{-z p_j}),$$

$$\mathbb{E}[\delta_A, \mathcal{P}_z, \mathcal{B}] = 2 \sum_{\substack{(i,j)\in\Sigma^2 \\ i<j}} \frac{p_i\, p_j}{{P_{[i,j]}}^2} \left[e^{-z P_{[i,j]}} - 1 + z P_{[i,j]} \right],$$

where $P_{[i,j]} = \sum_{k=i}^{j} p_k$ and $P_{[>j]} = \sum_{k>j} p_k$.

In each node, these computations are applied with

z replaced by zp_w and p_i replaced by $\dfrac{p_{w \cdot i}}{p_w}$

Mean trie costs in the Poisson Model

relative to the size of a trie, path length of an array-trie, path length of a list trie, path length of a bst-trie:

$$\widetilde{R}(z) = \sum_{w\in\Sigma^*} \left[1-(1+zp_w)e^{-zp_w}\right],$$

$$\widetilde{C}(z) = \sum_{w\in\Sigma^*} zp_w \left[1-e^{-zp_w}\right]$$

$$\widetilde{L}(z) = \sum_{w\in\Sigma^*} \sum_{i\in\Sigma} z\,P_{w\cdot[>i]}\,(1-e^{-zp_{w\cdot i}})$$

$$\widetilde{A}(z) = 2\sum_{w\in\Sigma^*} \sum_{\substack{(i,j)\in\Sigma^2\\ i<j}} \frac{p_{w\cdot i}\,p_{w\cdot j}}{{P_{w\cdot[i,j]}}^2}\left[e^{-zP_{w\cdot[i,j]}}-1+zP_{w\cdot[i,j]}\right],$$

where $P_{w\cdot[i,j]} = \sum_{k=i}^{j} p_{w\cdot k}$, and $P_{w\cdot[>j]} = \sum_{k>j} p_{w\cdot k}$.

Plan of the talk.

- Motivations : Realistic analyses of Sorting and Searching algorithms.
- A general model of source
- Description of the main results.
- Exact analysis in the Poisson model : the `Trie`
- Exact analysis in the Poisson model : `QuickSort` and `QuickSelect`.
- Exact analysis in the Bernoulli model
- Asymptotic analysis : different types of sources.

The mean number $\widetilde{S}(Z)$ of symbol comparisons for an algorithm $\mathcal{A}$ is

$$\widetilde{S}(Z) = \int_{\mathcal{T}} [\gamma(u,t) + 1]\, \widetilde{\pi}_Z(u,t)\, du\, dt$$

where $\mathcal{T} := \{(u,t),\ \ 0 \le u \le t \le 1\}$ is the unit triangle

$\gamma(u,t) :=$ coincidence between $M(u)$ and $M(t)$

$\widetilde{\pi}_Z(u,t)\, du\, dt :=$ Mean number of key-comparisons between $M(u')$ and $M(t')$ with $u' \in [u, u+du]$ and $t' \in [t-dt, t]$ performed by the algorithm $\mathcal{A}$.

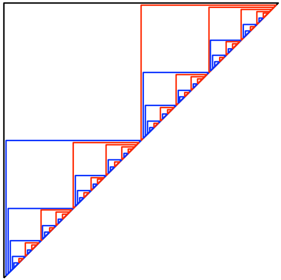
An (easy) alternative expression for $\widetilde{S}(Z)$

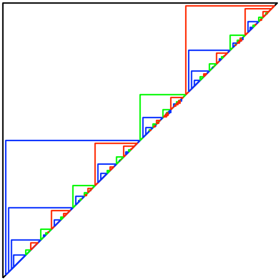
$$\widetilde{S}(Z) = \sum_{w \in \Sigma^\star} \int_{\mathcal{T}_w} \widetilde{\pi}_Z(u,t)\, du\, dt$$

It involves the fundamental triangles

and separates the rôles of the source and the algorithm.

Instances of fundamental triangles.





On the left:

memoryless source on $\{a, b\}$

$p_a = 1/2$, $p_b = 1/2$

On the right :

memoryless source on $\{a, b, c\}$

$p_a = 1/2$, $p_b = 1/6$, $p_c = 1/3$

Study of the key probability $\widetilde{\pi}_Z(u,t)$ of `QuickX` (`X= Sort` or `X= Quant`$_\alpha$.)

Related question : When does `QuickX` compare two keys $M(u)$ and $M(t)$?

In `QuickSort`, $M(u)$ and $M(t)$ are compared
iff the first pivot chosen in $\{M(v), v \in [u,t]\}$ is $M(u)$ or $M(t)$

In `QuickMin`, $M(u)$ and $M(t)$ are compared
iff the first pivot chosen in $\{M(v), v \in [0,t]\}$ is $M(u)$ or $M(t)$

In `QuickMax`, $M(u)$ and $M(t)$ are compared
iff the first pivot chosen in $\{M(v), v \in [u,1]\}$ is $M(u)$ or $M(t)$

And for `QuickQuant`$_\alpha$? Not so easy!

The idea is to compare `QuickQuant`
with a dual algorithm, the `QuickVal` algorithm.

A parenthesis – Presentation of QuickVal

The QuickVal algorithm is the dual algorithm of QuickSelect,

QuickVal (n, a, A). : returns the rank of the element a in $B = A \cup \{a\}$

$B := A \cup \{a\}$

QV (n, a, B);

QV (n, a, B).

Choose a pivot in B;

$(k, B_-, B_+) :=$ Partition(B);

If $a =$ pivot then QV $:= k$

else if $a <$ pivot then QV $:=$ QV $(k-1, a, B_-)$

else QV $:= k+$ QV $(n-k, a, B_+)$;

QuickVal$_\alpha$:= the algorithm where the key of interest is the word $M(\alpha)$

Comparison between $\texttt{QuickVal}_\alpha$ and $\texttt{QuickQuant}_\alpha$

$\texttt{QuickVal}_\alpha$:= the algorithm where the key of interest is the word $M(\alpha)$

There are two facts

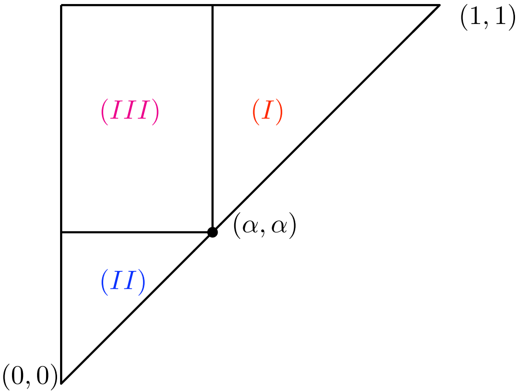
– Since the rank of $M(\alpha)$ amongst n keys is close to αn (for $n \to \infty$), the probabilistic behaviours of the two algorithms are close

– The $\texttt{QuickVal}_\alpha$ algorithm is easy to deal with since

$M(u)$ and $M(t)$ are compared in $\texttt{QuickVal}_\alpha$
iff the first pivot chosen in $\{M(v), v \in [x, y]\}$ is $M(u)$ or $M(t)$.

Here, the interval $[x, y]$ is the smallest interval that contains u, t and α.
this means : $x = \min(\alpha, u), \qquad y = \max(\alpha, t)$

The three domains for the definition of the interval $[x, y]$,
the smallest interval that contains u, t, α



$$[x(u,t), y(u,t)] := \begin{cases} [\alpha, t] & \text{if } u > \alpha \quad (I) \quad \sim \texttt{QuickMin} \\ [u, \alpha] & \text{if } t < \alpha \quad (II) \quad \sim \texttt{QuickMax} \\ [u, t] & \text{if } u < \alpha < t \quad (III) \quad \sim \texttt{QuickSort} \end{cases}$$

In summary, the algorithm $\texttt{QuickX}$ with X= $\texttt{Sort}$ or X= $\texttt{Val}_\alpha$,

compares two words $M(u)$ and $M(t)$

iff $M(u)$ or $M(t)$ is chosen as the first pivot in $\{M(v), v \in [x,y]\}$ with

$[x,y] = [u,t]$ ($\texttt{QuickSort}$), $\qquad [x,y] = [\min(\alpha,u), \max(\alpha,t)]$ ($\texttt{QuickVal}_\alpha$)

In the Poisson model, $\qquad \widetilde{\pi}_Z(u,t)\, du\, dt = Zdu \cdot Zdt \cdot \widetilde{\mathbb{E}}_Z \left[\dfrac{2}{2 + N_{[x,y]}}\right]$

$\widetilde{\pi}_Z(u,t) = 2\, Z^2\, f_1(Z(y-x)) \qquad$ with $\quad f_1(\theta) := \theta^{-2}\,[e^{-\theta} - 1 + \theta]$

With $f_0(\theta) = \theta(1 - e^{-\theta}), \quad f_1(\theta) := \theta^{-2}\,[e^{-\theta} - 1 + \theta]$,

Final expressions of the mean cost for $\texttt{Trie}$ and $\texttt{QuickX}$ in the $\mathcal{P}_Z$ model

$$\widetilde{C}(z) = \sum_{w \in \Sigma^\star} f_0(zp_w) \qquad \widetilde{S}(z) = 2z^2 \sum_{w \in \Sigma^\star} \int_{\mathcal{T}_w} f_1(z(y-x))\, dudt,$$

Plan of the talk.

– Motivations : Realistic analyses of Sorting and Searching algorithms.

– A general model of source

– Description of the main results.

– Exact analysis in the Poisson model : the `Trie`

– Exact analysis in the Poisson model : `QuickSort` and `QuickSelect`.

– Exact analysis in the Bernoulli model

– Asymptotic analysis : different types of sources.

Return to the model where the number n of keys is fixed.

Expanding f_0, f_1, $\quad f_0(\theta) = \theta[1 - e^{-\theta}], \qquad f_1(\theta) := \theta^{-2}\,[e^{-\theta} - 1 + \theta]$,

and using the transfer between the two models $\qquad \frac{S_n}{n!} = [Z^n]\left(e^Z \cdot \widetilde{S}_Z\right)$

there is an exact formula for S_n

$$S_n = 2\sum_{k=2}^{n}(-1)^k \binom{n}{k} \varpi(k)$$

which involves the series ϖ at integer values k.

The series $\varpi(s)$ is of Dirichlet type, and depends both

– on the algorithm (via the function f_0 or f_1 and interval $[x, y]$)

– on the source (via the fundamental triangles $\mathcal{T}_w$)

In the three cases, an exact formula for S_n ....

$$S_n = \sum_{k=2}^{n} (-1)^k \binom{n}{k} \varpi(k)$$

...which involves the series ϖ at integer values k.

For the mean path length (`Trie` or `BST`),

$\varpi(s)$ is closely related to the Dirichlet series of the probabilities,

$$\varpi_C(s) = s\Lambda(s) \qquad \varpi_B(s) = 2\frac{\Lambda(s)}{s(s-1)} \qquad \text{where} \quad \Lambda(s) := \sum_{w \in \Sigma^\star} p_w^s$$

For `QuickVal`, the expression is more involved,

$$\varpi_V(s) = 2 \sum_{w \in \Sigma^\star} \int_{\mathcal{T}_w} (y-x)^{s-2}\, du\, dt$$

Plan of the talk.

- Motivations : Realistic analyses of Sorting and Searching algorithms.
- A general model of source
- Description of the main results.
- Exact analysis in the Poisson model : the `Trie`
- Exact analysis in the Poisson model : `QuickSort` and `QuickSelect`.
- Exact analysis in the Bernoulli model
- Asymptotic analysis : different types of sources.

Asymptotic analysis.

Then, the residue formula transforms the sum into an integral:

$$S_n = \sum_{k=2}^{n} (-1)^k \binom{n}{k} \varpi(k) = \frac{1}{2i\pi} \int_{d-i\infty}^{d+i\infty} \varpi(s) \frac{n!\,(-1)^{n+1}}{s(s-1)\dots(s-n)} ds,$$

with $1 < d < 2$.

We shift the integral on the left,

and (usually) the first singularities occur at $\Re s = 1$.

What is the behaviour of $\varpi(s)$ near $\Re s = 1$?

We compare it to other Dirichlet series:

– For `Trie`, `BST`,

$\varpi_C(s), \varpi_B(s)$ are related to $\Lambda(s)$.

$$\Lambda(s) := \sum_{w \in \Sigma^\star} p_w^s,$$

$p_w = \Pr$ [a word begins with w],

– For `QuickVal`,

$\varpi_V(s)$ is related to $\Pi(s)$.

$$\Pi(s) = \sum_{k \geq 0} \pi_k^s.$$

$\pi_k = \sup\{p_w;\ w \in \Sigma^k\}$

Study of QuickVal and QuickQuant

A function is "tame" in a region $\mathcal{R}$

if it is analytic and of polynomial growth for $|s| \to \infty$

A source S is Π–tame if $\Pi(s)$ is tame on $\{\Re s > 1 - \delta\}$ with $\delta > 0$.

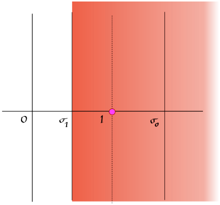
A sufficient condition is $\pi_k \leq Ak^{-\gamma}$ with $\gamma > 1$. Then $\delta = 1-(1/\gamma)$

Most of the "natural" sources are Π–tame !

In this case,

(1) $\varpi(s)$ is also tame in $\{\Re s > 1 - \delta\}$.

(2) The function $\alpha \mapsto \rho_{\mathcal{S}}(\alpha)$ is Hölder of exponent δ



(1) $\Rightarrow$ analysis of `QuickVal`

(2) $\Rightarrow$ analysis of `QuickQuant`

A nice expression for $\rho_{\mathcal{S}}(\alpha) = \sum_{w \in \Sigma^\star} \int_{\mathcal{T}_w} [\max(\alpha, t) - \min(\alpha, u)]^{-1} du dt$

Study of the mean path length of `Trie` and `BST`

$$\varpi_T(s) = s\Lambda(s), \qquad \varpi_B(s) = 2\frac{\Lambda(s)}{s(s-1)} \quad \text{where} \quad \Lambda(s) := \sum_{w\in\Sigma^\star} p_w^s$$

For any (natural) source, $\Lambda(s)$ has a singularity at $s = 1$.

A source is Λ–tame if

(1) the dominant singularity of $\Lambda(s)$ is located at $s = 1$,

this is a simple pôle, whose residue equals $1/h_{\mathcal{S}}$.

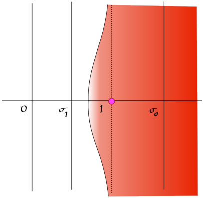
In this case, there is, at $s = 1$

a double pôle for $\dfrac{\varpi_C(s)}{s-1}$, a triple pôle for $\dfrac{\varpi_B(s)}{s-1}$

(2) $\Lambda(s)$ is tame on the left of the line $\Re s = 1$

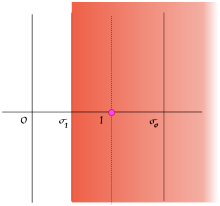
(useful for shifting on the left...)

Different possible regions on the left of $\Re s = 1$ where $\Lambda(s)$ is tame.



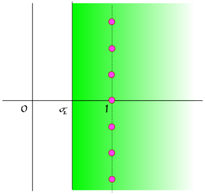
Situation 1

Hyperbolic region



Situation 2

Vertical strip



Situation 3

Vertical strip with holes

For which (simple) sources do these different situations occur?

For memoryless sources relative to $\mathfrak{P} = (p_1, p_2, \ldots, p_r)$

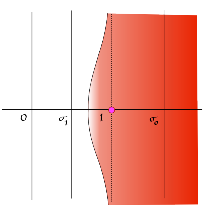
– S2 is impossible

– S3 occurs when all the ratios $\log p_i / \log p_j$ are rational

– S1 with a frontier of the form $\sigma = 1 - A/t^{\alpha}$ occurs

if there exists a ratio $\log p_i / \log p_j$ which badly approximable by rationals.

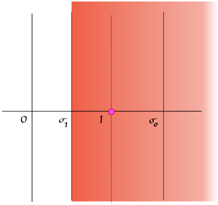
Different possible regions on the left of $\Re s = 1$ where $\Lambda(s)$ is tame.



Situation 1

Hyperbolic region

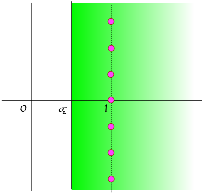
Arithmetic condition



Situation 2

Vertical strip

Geometric condition



Situation 3

Vertical strip with holes

Periodicity condition

For which sources do these different situations occur?

For dynamical sources, we provide sufficient conditions under which these behaviours hold.

– S3 never occurs except if the source is conjugated to a simple source.

– S2 occurs when all the branches are not too often of the same geometric form

– S1 occurs if a extension of the following condition holds:

there exists a ratio $\log p_i / \log p_j$ which badly approximable by rationals.

Conclusions.

— For any Λ–tame source, the mean path-lengths of `Trie` and `BST` are

$$C(n) \sim \frac{1}{h_{\mathcal{S}}}\, n \log n \quad (\texttt{Trie}), \qquad B(n) \sim \frac{1}{h_{\mathcal{S}}}\, n \log^2 n \quad (\texttt{BST})$$

— It is easy to adapt our results to the intermittent sources, which emits "long" sequences of the same symbols. In this case,

$$C(n) = \Theta(n \log^2 n). \quad (\texttt{Trie}) \qquad B(n) = \Theta(n \log^3 n), \quad (\texttt{BST})$$

— For any reasonable source, $Q(n) = \Theta(n)$ (`QuickQuant`).

Long term research projects...

— Revisit the complexity results of the main classical algorithms,
and take into account the number of symbol-comparisons...
instead of the number of key-comparisons.

— Provide a sharp "analytic" classification of sources:
Transfer probabilistic properties of sources into analytical properties of $\Lambda(s)$.