TREES and SOURCES

How to sort n words?

Dedicated to Philippe for his 60th Birthday

Brigitte Vallée (CNRS and Université de Caen, France)

How to sort n words?

– Which sorting methods? based on which underlying data structures?

– Which words? emitted by which source?

How to sort n words?

– Which sorting methods? based on which underlying data structures?

– Which words? emitted by which source?

We focus on two main data structures,

– the Trie, underlying the `RadixSort` algorithm,

– the Binary Search Tree, underlying the `QuickSort` algorithm,

How to sort n words?

– Which sorting methods? based on which underlying data structures?

– Which words? emitted by which source?

We focus on two main data structures,

– the Trie, underlying the `RadixSort` algorithm,

– the Binary Search Tree, underlying the `QuickSort` algorithm,

built on words independently emitted by the same general tamed source....

How to sort n words?

– Which sorting methods? based on which underlying data structures?
– Which words? emitted by which source?

We focus on two main data structures,
– the Trie, underlying the `RadixSort` algorithm,
– the Binary Search Tree, underlying the `QuickSort` algorithm,

built on words independently emitted by the same general tamed source....

We also describe the particular case of the continued fraction source.
How to sort numbers given by their continued fraction expansions ?
The tameness of the CF–source is closely related to the Riemann hypothesis.

These results are obtained in joint works with Philippe
during these last ten years
These joint works have been very fruitful for my own research

These results are obtained in joint works with Philippe
during these last ten years
These joint works have been very fruitful for my own research

1998. We analyze the trie on the CF-source (Clément, Flajolet, V.)

These results are obtained in joint works with Philippe
during these last ten years
These joint works have been very fruitful for my own research

1998. We analyze the trie on the CF-source (Clément, Flajolet, V.)
1999. I have the idea of modeling sources via dynamical systems.

These results are obtained in joint works with Philippe
during these last ten years
These joint works have been very fruitful for my own research

1998. We analyze the trie on the CF-source (Clément, Flajolet, V.)
1999. I have the idea of modeling sources via dynamical systems.
2001. We analyze tries or ternary search tries built on a general source...
provided that the source be tamed (Clément, Flajolet, V.)

These results are obtained in joint works with Philippe
during these last ten years
These joint works have been very fruitful for my own research

1998. We analyze the trie on the CF-source (Clément, Flajolet, V.)
1999. I have the idea of modeling sources via dynamical systems.
2001. We analyze tries or ternary search tries built on a general source... provided that the source be tamed (Clément, Flajolet, V.)
2004-2006. I adapt Dolgopyat's works and exhibit geometric conditions for a dynamical source to be tamed (Baladi, Cesaratto, V.)

These results are obtained in joint works with Philippe
during these last ten years
These joint works have been very fruitful for my own research

1998. We analyze the trie on the CF-source (Clément, Flajolet, V.)
1999. I have the idea of modeling sources via dynamical systems.
2001. We analyze tries or ternary search tries built on a general source... provided that the source be tamed (Clément, Flajolet, V.)
2004-2006. I adapt Dolgopyat's works and exhibit geometric conditions for a dynamical source to be tamed (Baladi, Cesaratto, V.)
2008. We analyze binary search trees built on a general source... provided that the source be tamed... (Clément, Flajolet, Fill, V.)

These results are obtained in joint works with Philippe
during these last ten years
These joint works have been very fruitful for my own research

1998. We analyze the trie on the CF-source (Clément, Flajolet, V.)
1999. I have the idea of modeling sources via dynamical systems.
2001. We analyze tries or ternary search tries built on a general source... provided that the source be tamed (Clément, Flajolet, V.)
2004-2006. I adapt Dolgopyat's works and exhibit geometric conditions for a dynamical source to be tamed (Baladi, Cesaratto, V.)
2008. We analyze binary search trees built on a general source... provided that the source be tamed... (Clément, Flajolet, Fill, V.)
2009. I provide more general sufficient conditions for a dynamical source to be tamed: work in progress (Cesaratto, Roux, V.)

Plan of the talk.

– The data structures, the Trie and the BST
– The main result
– The model of sources.
– The main steps of the method.
– What is a tamed source?
– The particular case of the Continued Fraction Source.

Plan of the talk.

– The data structures, the Trie and the BST
– The main result
– The model of sources.
– The main steps of the method.
– What is a tamed source?
– The particular case of the Continued Fraction Source.

The classical framework for sorting.

The main sorting algorithms or searching algorithms

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

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 classical framework for sorting.

The main sorting algorithms or searching algorithms

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

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.

A more realistic framework for sorting.

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

$\Sigma^\infty = \{$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 classical framework for sorting.

The main sorting algorithms or searching algorithms

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

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.

A more realistic framework for sorting.

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

$\Sigma^\infty = \{$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

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

Here, we perform a realistic analysis of the `QuickSort` algorithm and its underlying data structure, the Binary Search Tree (BST), with respect to the number of symbol–comparisons

An initial question asked by Sedgewick in 2000, in order to compare with algorithms of type `Radix-Sort` based on Tries.

Here, we perform a realistic analysis of the `QuickSort` algorithm and its underlying data structure, the Binary Search Tree (BST), with respect to the number of symbol–comparisons

An initial question asked by Sedgewick in 2000, in order to compare with algorithms of type `Radix-Sort` based on Tries.

A comparison between three mean path lengths, with n data.

– The mean classical path length K_n of the the BST.

– The mean realistic path length B_n of the the BST

– The mean path length T_n of the trie

Here, we perform a realistic analysis of the `QuickSort` algorithm and its underlying data structure, the Binary Search Tree (BST), with respect to the number of symbol–comparisons

An initial question asked by Sedgewick in 2000, in order to compare with algorithms of type `Radix-Sort` based on Tries.

A comparison between three mean path lengths, with n data.

– The mean classical path length K_n of the the BST.
– The mean realistic path length B_n of the the BST
– The mean path length T_n of the trie

An example.

Sixteen words of length 12 ....

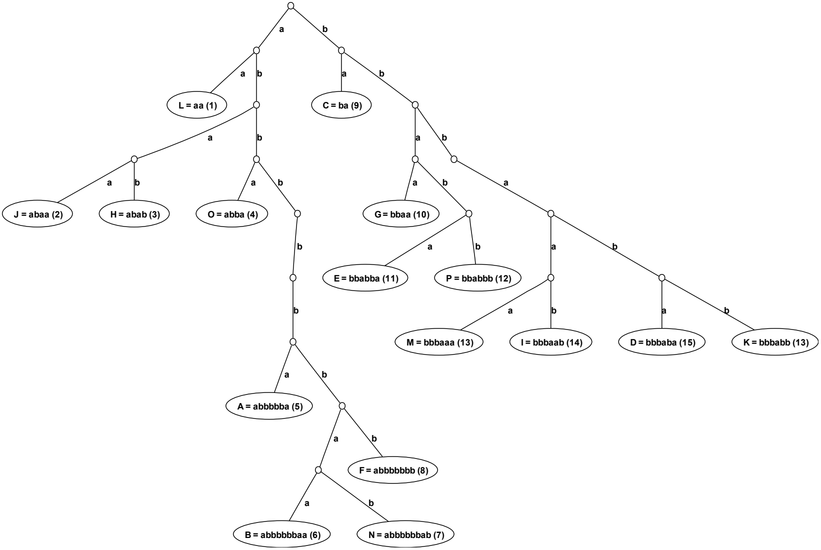
drawn from the memoryless source $p(a) = 1/3, p(b) = 2/3$.....

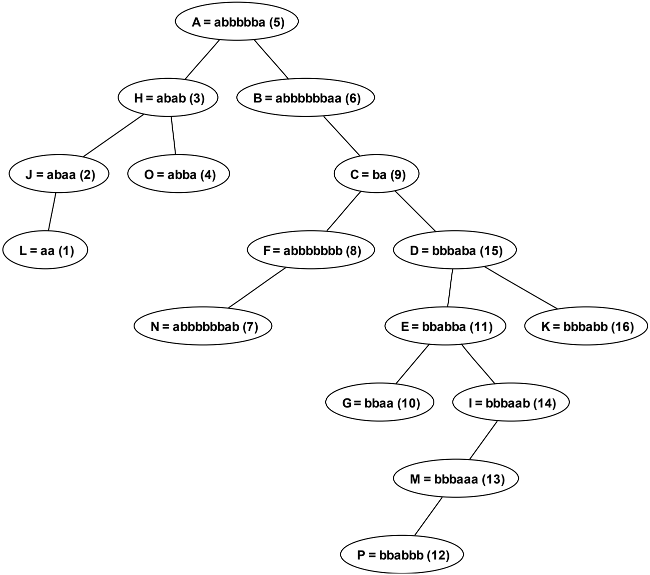
Observe the trie and the BST built on this sequence of words...

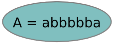
A = abbbbbaaabab   B = abbbbbbaabaa   C = baabbbabbbba   D = bbbababbbaab   E = bbabbaababbb

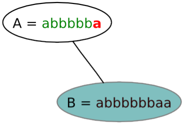
F = abbbbbbbbabb   G = bbaabbabbaba   H = ababbbabbbab   I = bbbaabbbbbbb   J = abaabbbbaabb

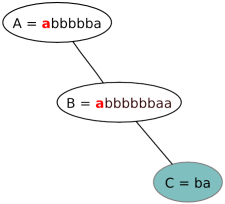
= bbbabbbbbbaa   L = aaaabbabaaba   M = bbbaaabbbbbb   N = abbbbbbabbaa   O = abbababababbbb   P = bbabbbaaa

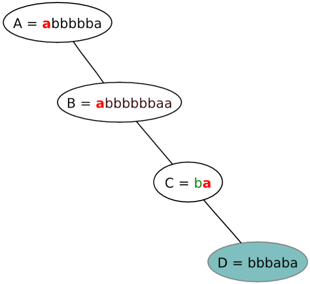


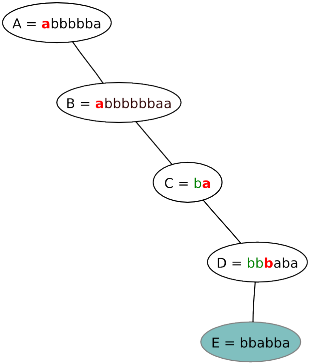

A = abbbbba (5)
H = abab (3)
B = abbbbbbaa (6)
J = abaa (2)
O = abba (4)
C = ba (9)
L = aa (1)
F = abbbbbbb (8)
D = bbbaba (15)
N = abbbbbbab (7)
E = bbabba (11)
K = bbbabb (16)
G = bbaa (10)
I = bbbaab (14)
M = bbbaaa (13)
P = bbabbb (12)

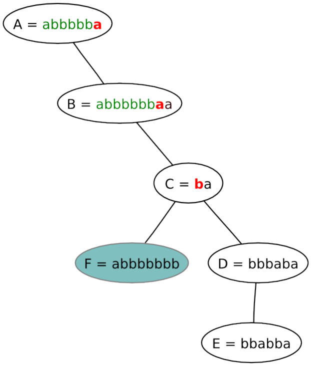

A = abbbbba


A = abbbbba
B = abbbbbbaa

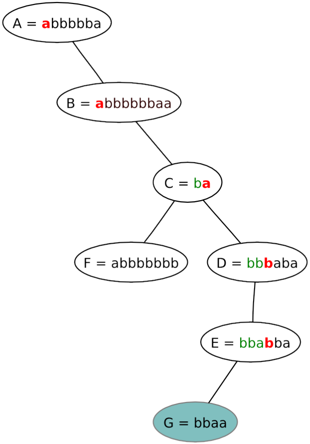

A = abbbbba
B = abbbbbbaa
C = ba

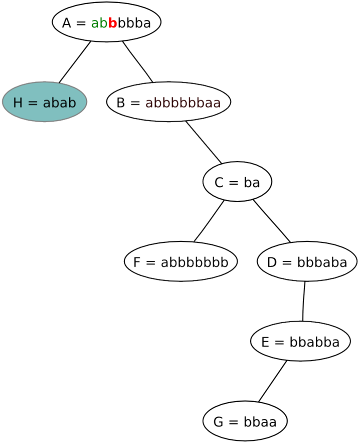

A = abbbbba
B = abbbbbbaa
C = ba
D = bbbaba

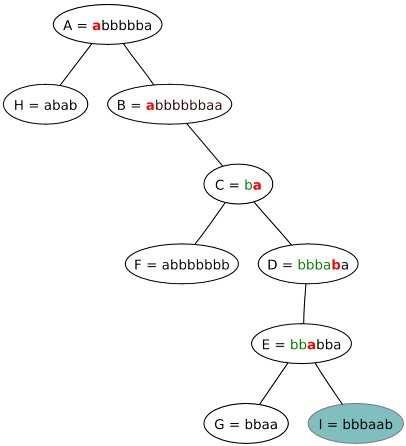

A = abbbbba
B = abbbbbbaa
C = ba
D = bbbaba
E = bbabba



A = abbbbba
B = abbbbbbaa
C = ba
F = abbbbbbb
D = bbbaba
E = bbabba


A = abbbbba
B = abbbbbbaa
C = ba
F = abbbbbbb
D = bbbaba
E = bbabba
G = bbaa


A = abbbbba
H = abab
B = abbbbbbaa
C = ba
F = abbbbbbb
D = bbbaba
E = bbabba
G = bbaa


A = abbbbba
H = abab
B = abbbbbbaa
C = ba
F = abbbbbbb
D = bbbaba
E = bbabba
G = bbaa
I = bbbaab

Plan of the talk.

– The data structures, the Trie and the BST

– The main result

– The model of sources.

– The main steps of the method.

– What is a tamed source?

– The particular case of the Continued Fraction Source.

For n words independently drawn from the same tamed general source,
– the mean path length T_n of a trie,
– the mean symbol–path length B_n of a BST
= the mean number of symbol comparisons in `QuickSort`

satisfy $T_n \sim \frac{1}{h_{\mathcal{S}}} n \log n, \qquad B_n \sim \frac{1}{h_{\mathcal{S}}} n \log^2 n.$

They 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.

Same results previously obtained for tries and BST on particular sources

For n words independently drawn from the same tamed general source,
– the mean path length T_n of a trie,
– the mean symbol–path length B_n of a BST
= the mean number of symbol comparisons in `QuickSort`

satisfy $$T_n \sim \frac{1}{h_{\mathcal{S}}}\, n\, \log n, \qquad B_n \sim \frac{1}{h_{\mathcal{S}}}\, n\, \log^2 n.$$

They 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.

Compared to the mean key–path length K_n of the BST, $K_n \sim 2n \log n$,
B_n has an extra factor $1/(2h_{\mathcal{S}}) \log n$

Compared to the mean path length T_n of the trie, B_n has an extra factor $\log n$

Plan of the talk.

– The data structures, the Trie and the BST

– The main result

– The model of sources.

– The main steps of the method.

– What is a tamed source?

– The particular case of the Continued Fraction Source.

The (general) model of source.

Source. A general source $\mathcal{S}$ produces words on an alphabet Σ.

To $u \in \mathcal{I} := [0,1]$ it associates a word $M(u) \in \Sigma^\infty$.

The lexicographic order on Σ^∞ is compatible with the order on $\mathcal{I}$.

The (general) model of source.

Source. A general source $\mathcal{S}$ produces words on an alphabet Σ.
To $u \in \mathcal{I} := [0, 1]$ it associates a word $M(u) \in \Sigma^\infty$.
The lexicographic order on Σ^∞ is compatible with the order on $\mathcal{I}$.

For any source $\mathcal{S}$, for any prefix $w \in \Sigma^\star$,
the reals u for which the word $M(u)$ begins with w form an interval,
denoted by $\mathcal{I}_w$, called the fundamental interval relative to the prefix w.

The (general) model of source.

Source. A general source $\mathcal{S}$ produces words on an alphabet Σ.
To $u \in \mathcal{I} := [0, 1]$ it associates a word $M(u) \in \Sigma^\infty$.
The lexicographic order on Σ^∞ is compatible with the order on $\mathcal{I}$.

For any source $\mathcal{S}$, for any prefix $w \in \Sigma^\star$,
the reals u for which the word $M(u)$ begins with w form an interval,
denoted by $\mathcal{I}_w$, called the fundamental interval relative to the prefix w.

The measure of the interval $\mathcal{I}_w$ is the probability that $M(u)$ begins with w,
p_w, called the fundamental probability of the prefix w.

The (general) model of source.

Source. A general source $\mathcal{S}$ produces words on an alphabet Σ.
To $u \in \mathcal{I} := [0, 1]$ it associates a word $M(u) \in \Sigma^\infty$.
The lexicographic order on Σ^∞ is compatible with the order on $\mathcal{I}$.

For any source $\mathcal{S}$, for any prefix $w \in \Sigma^\star$,
the reals u for which the word $M(u)$ begins with w form an interval,
denoted by $\mathcal{I}_w$, called the fundamental interval relative to the prefix w.

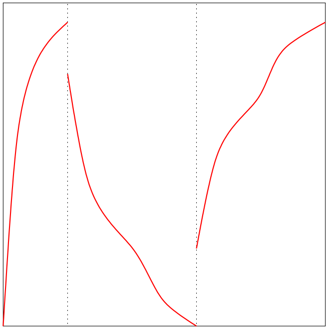
The measure of the interval $\mathcal{I}_w$ is the probability that $M(u)$ begins with w,
p_w, called the fundamental probability of the prefix w.

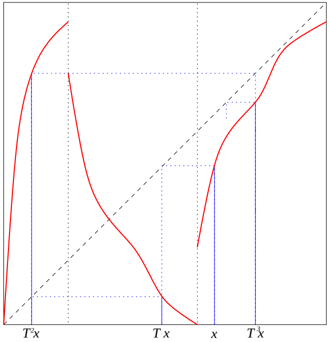
A main (analytical) object:
the Dirichlet series of fundamental probabilities,

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

Natural instances of sources: Dynamical sources

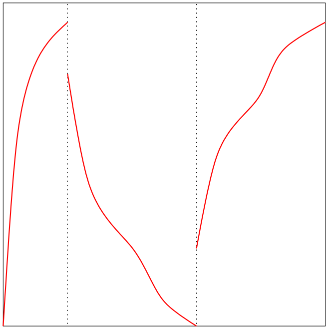
With a shift map $T : \mathcal{I} \to \mathcal{I}$ and an encoding map $\tau : \mathcal{I} \to \Sigma$, the emitted word is $M(x) = (\tau x, \tau T x, \tau T^2 x, \ldots \tau T^k x, \ldots)$

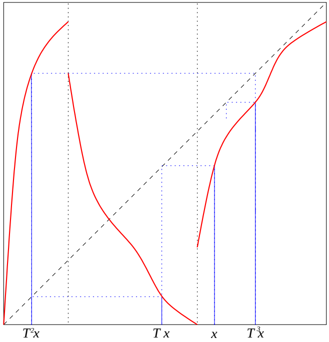




Natural instances of sources: Dynamical sources

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

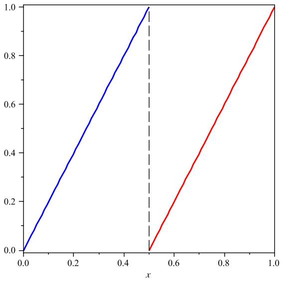


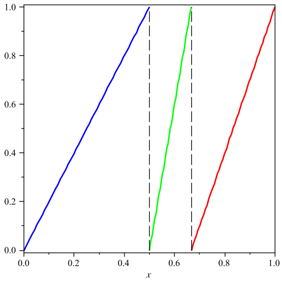


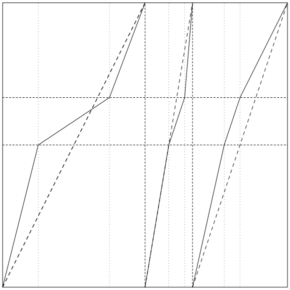
A dynamical system, with $\Sigma = \{a, b, c\}$ and a word $M(x) = (c, b, a, c \ldots)$.

Memoryless sources or Markov chains.

= Dynamical sources with affine branches....







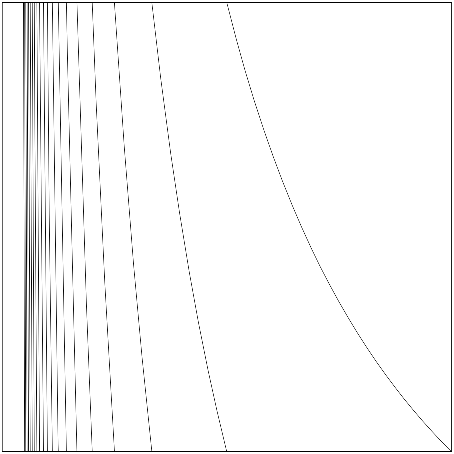
The dynamical framework leads to more general sources.

The curvature of branches entails correlation between symbols

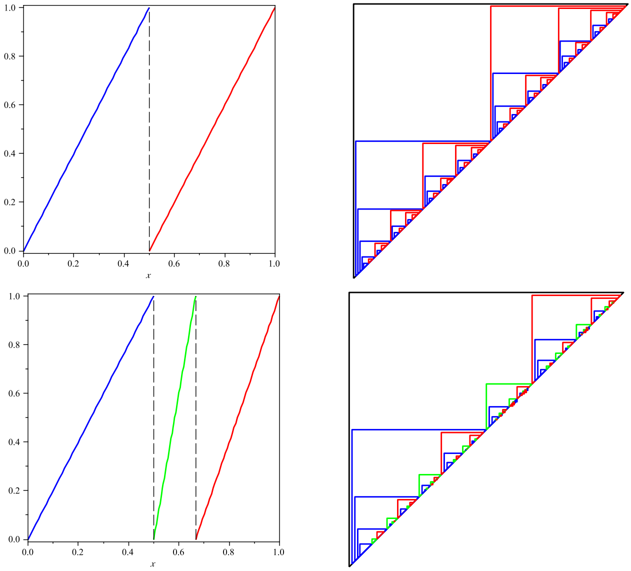
The dynamical framework leads to more general sources.

The curvature of branches entails correlation between symbols

Example : the Continued Fraction source



Fundamental intervals and fundamental triangles.



Plan of the talk.

– The data structures, the Trie and the BST

– The main result

– The model of sources.

– The main steps of the method.

– What is a tamed source?

– The particular case of the Continued Fraction Source.

Three main steps for the analysis of the path length S_n

Three main steps for the analysis of the path length S_n

(A) The Poisson model $\mathcal{P}_Z$ does not deal with a fixed number n of keys. The number N of keys is now a random variable which follows a Poisson law of parameter Z.

We first obtain nice expressions for $\widetilde{S}_Z$ ....

Three main steps for the analysis of the path length S_n

(A) The Poisson model $\mathcal{P}_Z$ does not deal with a fixed number n of keys. The number N of keys is now a random variable which follows a Poisson law of parameter Z.

We first obtain nice expressions for $\widetilde{S}_Z$ ....

(B) It is now possible to returm 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...

Three main steps for the analysis of the path length S_n

(A) The Poisson model $\mathcal{P}_Z$ does not deal with a fixed number n of keys. The number N of keys is now a random variable which follows a Poisson law of parameter Z.

We first obtain nice expressions for $\widetilde{S}_Z$ ....

(B) It is now possible to returm 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...

(C) Then, the Rice formula provides the asymptotics of S_n ($n \to \infty$), as soon as the source is "tamed".

(A) **Dealing with the Poisson Model.**

In the $\mathcal{P}_Z$ model, the number N of keys follows the Poisson law

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

the mean number $\widetilde{S}(Z)$ of symbol comparisons for building the structure is expressed as:

- a sum over the set $\Sigma^\star$ of all possible finite prefixes,
- each term $\widetilde{S}_w(Z)$ dealing with a prefix w.

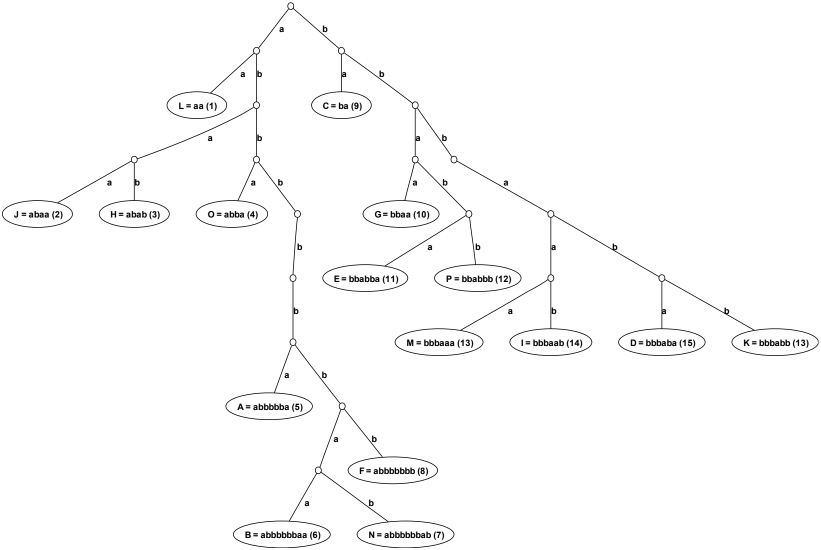
Trie. The contribution $\widetilde{T}_w(Z)$ of prefix w to the path length is

$$\widetilde{T}_w(Z) = \mathbb{E}[\underline{N}_w] = Zp_w[1 - e^{-Zp_w}],$$

where N_w is the number of words that begin with prefix w,

$$\underline{N}_w = \mathbf{1}_{[N_w \geq 2]} \cdot N_w$$

N_w follows a Poisson law of parameter Zp_w.



a
b
a
b
a
b
L = aa (1)
C = ba (9)
a
b
a
b
J = abaa (2)
H = abab (3)
O = abba (4)
G = bbaa (10)
a
b
a
b
a
a
b
E = bbabba (11)
P = bbabbb (12)
a
b
a
b
M = bbbaaa (13)
I = bbbaab (14)
D = bbbaba (15)
K = bbbabb (13)
b
b
a
b
A = abbbbba (5)
a
b
F = abbbbbbb (8)
a
b
B = abbbbbbaa (6)
N = abbbbbbab (7)

BST. The mean number of symbol–comparisons is

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

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

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

$\pi(u,t)\, du\, dt$:= Mean number of key-comparisons between $M(u')$ and $M(t')$ with $u' \in [u, u+du]$ and $t' \in [t-dt, t]$.

BST. The mean number of symbol–comparisons is

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

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

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

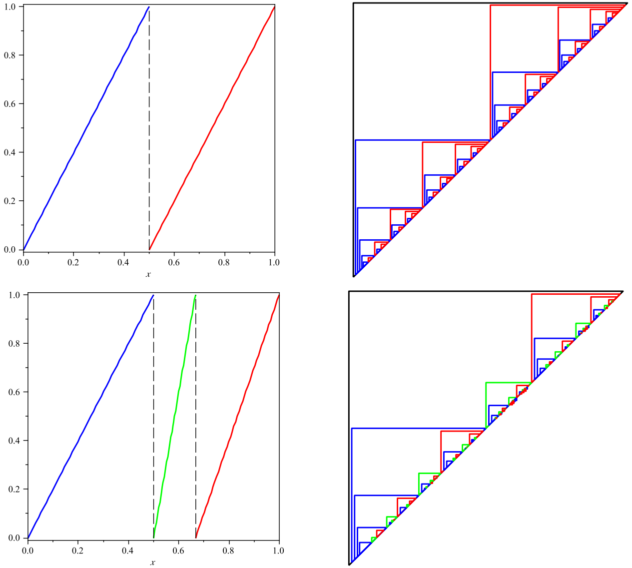
$\pi(u,t)\, du\, dt :=$ Mean number of key-comparisons between $M(u')$ and $M(t')$ with $u' \in [u, u+du]$ and $t' \in [t-dt, t]$.

(a) An (easy) alternative expression for

$$\widetilde{B}(Z) = \int_{\mathcal{T}} [\gamma(u,t)+1]\, \pi(u,t)\, du\, dt = \sum_{w \in \Sigma^\star} \int_{\mathcal{T}_w} \pi(u,t)\, du\, dt.$$

which involves the fundamental triangles
and separates the rôles of the source and the algorithm.

Fundamental intervals and fundamental triangles.



BST. The mean number of symbol–comparisons is

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

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

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

$\pi(u,t)\, du\, dt :=$ Mean number of key-comparisons between $M(u')$ and $M(t')$ with $u' \in [u, u+du]$ and $t' \in [t-dt, t]$.

(b) A nice expression for $\pi(u,t)$:

$M(u)$ and $M(t)$ are compared in `QuickSort`

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

$$\pi(u,t)dudt = Zdu \cdot Zdt \cdot \mathbb{E}\left[\frac{2}{2+N_{[u,t]}}\right] = (Z^2 dudt) \cdot 2 f_1(Z(t-u))$$

where $N_{[u,t]}$ is the number of words $M(x)$ with $x \in [u+du, t-dt]$, (which follows a Poisson law of parameter $Z(t-u)$)

and $f_1(\theta) := \theta^{-2}[e^{-\theta} - 1 + \theta]$.

BST. The mean number of symbol–comparisons is

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

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

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

$\pi(u,t)\, du\, dt :=$ Mean number of key-comparisons between $M(u')$ and $M(t')$ with $u' \in [u, u+du]$ and $t' \in [t-dt, t]$.

With (a) and (b), it is equal to

$$\widetilde{B}(Z) = 2Z^2 \sum_{w \in \Sigma^\star} \int_{\mathcal{T}_w} f_1(Z(t-u)) du dt$$

and involves

– a sum taken over all the prefixes $w \in \Sigma^\star$,

– the fundamental triangles $\mathcal{T}_w$,

– the function $f_1(\theta) := \theta^{-2}[e^{-\theta} - 1 + \theta]$.

(A) **Dealing with the Poisson Model.**

In the $\mathcal{P}_Z$ model, the number N of keys follows the Poisson law

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

the mean number $\widetilde{S}(Z)$ of symbol comparisons for building the structure is expressed as:

– a sum over the set $\Sigma^\star$ of all possible finite prefixes,

– each term $\widetilde{S}_w(Z)$ dealing with a prefix w.

Both for the Trie and the BST:

$$\widetilde{T}(Z) = \sum_{w\in\Sigma^\star} f_0(Zp_w), \qquad \widetilde{B}(Z) = 2Z^2 \sum_{w\in\Sigma^\star} \int_{\mathcal{T}_w} f_1(Z(t-u))dudt$$

with f_0, f_1 of exponential type...

(B) Return to the model where n is fixed.

With the expansions of f_0, f_1,

$$\widetilde{S}(Z) = \sum_{k=2}^{\infty} (-1)^k \varpi(-k) \frac{Z^k}{k!},$$

is expressed with a series $\varpi(s)$ of Dirichlet type,

which depends both on the data structure and the source.

(B) **Return to the model where n is fixed.**

With the expansions of f_0, f_1,

$$\widetilde{S}(Z) = \sum_{k=2}^{\infty} (-1)^k \varpi(-k) \frac{Z^k}{k!},$$

is expressed with a series $\varpi(s)$ of Dirichlet type,

which depends both on the data structure and the source.

Series ϖ_T, ϖ_B are related to the Dirichlet series $\Lambda(s)$ of probabilities

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

(B) **Return to the model where n is fixed.**

With the expansions of f_0, f_1,

$$\widetilde{S}(Z) = \sum_{k=2}^{\infty} (-1)^k \varpi(-k) \frac{Z^k}{k!},$$

is expressed with a series $\varpi(s)$ of Dirichlet type,

which depends both on the data structure and the source.

Series ϖ_T, ϖ_B are related to the Dirichlet series $\Lambda(s)$ of probabilities

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

Since $\dfrac{S_n}{n!} = [Z^n] \left(e^Z \cdot \widetilde{S}(Z) \right)$, there are exact formulae for T_n and B_n

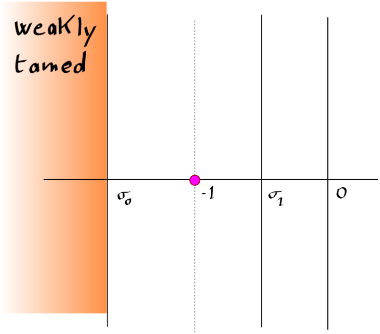
$$T_n = \sum_{k=2}^{n} (-1)^k \binom{n}{k} k\Lambda(-k) \qquad B_n = 2\sum_{k=2}^{n} (-1)^k \binom{n}{k} \frac{\Lambda(-k)}{k(k-1)}.$$

(C) **Using Rice formula**

As soon as $\varpi(s)$ is "weakly tamed" in $\Re(s) < \sigma_0$ with $\sigma_0 > -2$,
the residue formula transforms the sum into an integral:

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

with $-2 < d < \min(-1, \sigma_0)$.



(C) **Using Rice formula**

As soon as $\varpi(s)$ is "weakly tamed" in $\Re(s) < \sigma_0$ with $\sigma_0 > -2$,
the residue formula transforms the sum into an integral:

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

with $-2 < d < \min(-1, \sigma_0)$.

Where are the singularities ?

Recall: $\varpi_B(s) = 2\dfrac{\Lambda(s)}{s(s+1)}$, or $\varpi_T(s) = -s\Lambda(s)$,

where $\Lambda(s) := \sum_{w \in \Sigma^\star} p_w^{-s}$ has always a singularity at $s = -1$.

What type of singularity? Is it the dominant singularity?

– The data structures, the Trie and the BST
– The main result
– The model of sources.
– The main steps of the method.
– What is a tamed source?
– The particular case of the Continued Fraction Source.

What can be expected about $\Lambda(s)$?

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

What can be expected about $\Lambda(s)$?

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

— For a tamed source $\mathcal{S}$, the dominant singularity of $\Lambda(s)$ is located at $s = -1$, this is a simple pôle, whose residue equals $1/h_{\mathcal{S}}$.

What can be expected about $\Lambda(s)$?

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

— For a tamed source $\mathcal{S}$, 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 a double pôle at $s = -1$ for $$\frac{\varpi_T(s)}{s+1} = \frac{-s\Lambda(s)}{s+1}$$

and $$\frac{\varpi_T(s)}{s+1} \sim \frac{1}{h_{\mathcal{S}}} \frac{1}{(s+1)^2} \qquad s \to -1$$

What can be expected about $\Lambda(s)$?

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

— For a tamed source $\mathcal{S}$, 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 a double pôle at $s=-1$ for $\dfrac{\varpi_T(s)}{s+1} = \dfrac{-s\Lambda(s)}{s+1}$

and $$\frac{\varpi_T(s)}{s+1} \sim \frac{1}{h_{\mathcal{S}}}\frac{1}{(s+1)^2} \qquad s \to -1$$

—In this case, there is a triple pôle at $s=-1$ for $\dfrac{\varpi_B(s)}{s+1} = 2\dfrac{\Lambda(s)}{s(s+1)^2}$

and $$\frac{\varpi_B(s)}{s+1} \sim \frac{2}{h_{\mathcal{S}}}\frac{1}{(s+1)^3} \qquad s \to -1$$

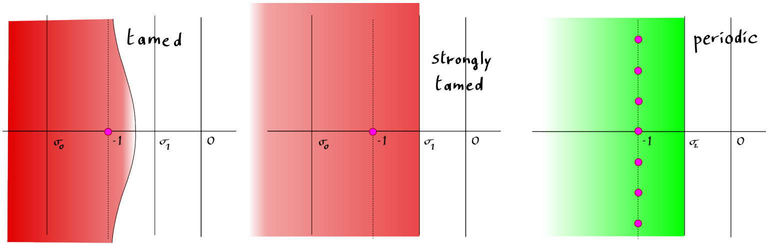
For shifting the integral to the right, past... $d = -1$,

other properties of $\Lambda(s)$ are needed on $\Re s \geq -1$, –more subtle–

Different behaviours of $\Lambda(s)$ for $\Re s \geq -1$ where one can past $d = -1$...

For shifting the integral to the right, past... $d = -1$,
other properties of $\Lambda(s)$ are needed on $\Re s \geq -1$, –more subtle–

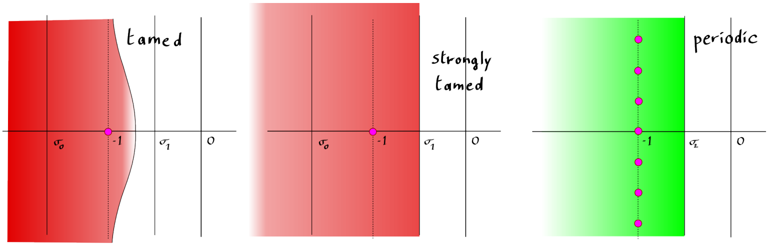
Different behaviours of $\Lambda(s)$ for $\Re s \geq -1$ where one can past $d = -1$...



In colored domains, $\Lambda(s)$ is meromorphic and of polynomial growth for $|s| \to \infty$.

For shifting the integral to the right, past... $d = -1$,
other properties of $\Lambda(s)$ are needed on $\Re s \geq -1$, –more subtle–

Different behaviours of $\Lambda(s)$ for $\Re s \geq -1$ where one can past $d = -1$...



In colored domains, $\Lambda(s)$ is meromorphic and of polynomial growth for $|s| \to \infty$.

For dynamical sources, we provide sufficient conditions
(of geometric or arithmetic type), under which these behaviours hold.

For a memoryless source, they depend on the approximability of ratios $\log p_i / \log p_j$

Plan of the talk.

– The data structures, the Trie and the BST
– The main result
– The model of sources.
– The main steps of the method.
– What is a tamed source?
– The particular case of the Continued Fraction Source.

The Continued Fraction Source

The Dirichlet series of fundamental probabilities satisfies

$$\Lambda(-s) = 2^{-s} + \left[2^{s-1} - 1\right] \frac{\zeta(s)^2}{\zeta(2s)} + \frac{2^s}{\zeta(2s)} \zeta^{-+}(s)$$

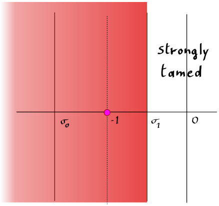
where the alternating zeta function $\zeta^{-+}(s)$ is defined as

$$\zeta^{-+}(s) := \sum_{n=1}^{+\infty} \frac{(-1)^n}{n^s} \sum_{q=1}^{n-1} \frac{1}{q^s}.$$

It is an entire function.

Then, the continued fraction source is strongly tamed, with an abscissa σ_1 related to s for which $\zeta(2s) = 0$.

If the Riemann hypothesis is true, one can choose $\sigma_1 = -1/4$.



Conclusions.

— Our methods apply to the mean number of symbol-comparisons in `QuickSelMin` and `QuickSelRand` (Clément, Fill, Flajolet, V. 08).
It is sufficient that the source be weakly tamed.

Conclusions.

— Our methods apply to the mean number of symbol-comparisons in `QuickSelMin` and `QuickSelRand` (Clément, Fill, Flajolet, V. 08).
It is sufficient that the source be weakly tamed.

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

$$S_n = \Theta(n \log^3 n), \qquad T_n = \Theta(n \log^2 n).$$

Conclusions.

— Our methods apply to the mean number of symbol-comparisons in `QuickSelMin` and `QuickSelRand` (Clément, Fill, Flajolet, V. 08).
It is sufficient that the source be weakly tamed.

— It is easy to adapt our results to the intermittent sources, which emit “long” sequences of the same symbol. In this case,

$$S_n = \Theta(n \log^3 n), \qquad T_n = \Theta(n \log^2 n).$$

— What about the distribution of the average search cost in a BST?

Is it asymptotically normal?

We know that this is true if one counts the number of key–comparisons.
We also know that, for a tamed source, the average depth of a trie is asymptotically normal (Cesaratto-Vallée, 2007).