The Number of Symbol Comparisons in QuickSort & QuickSelect

I. Overview ~~~ *Philippe Flajolet*

II. Average-Case Analysis ~~~ *Brigitte Vallée*

III. Distributions ~~~ *Jim Fill*

- 1. Algorithms & analysis
- 2. Cost measures
- 3. Sources (data model)
- 4. Results: average-case & distributional

1.QuickSort & QuickSelect

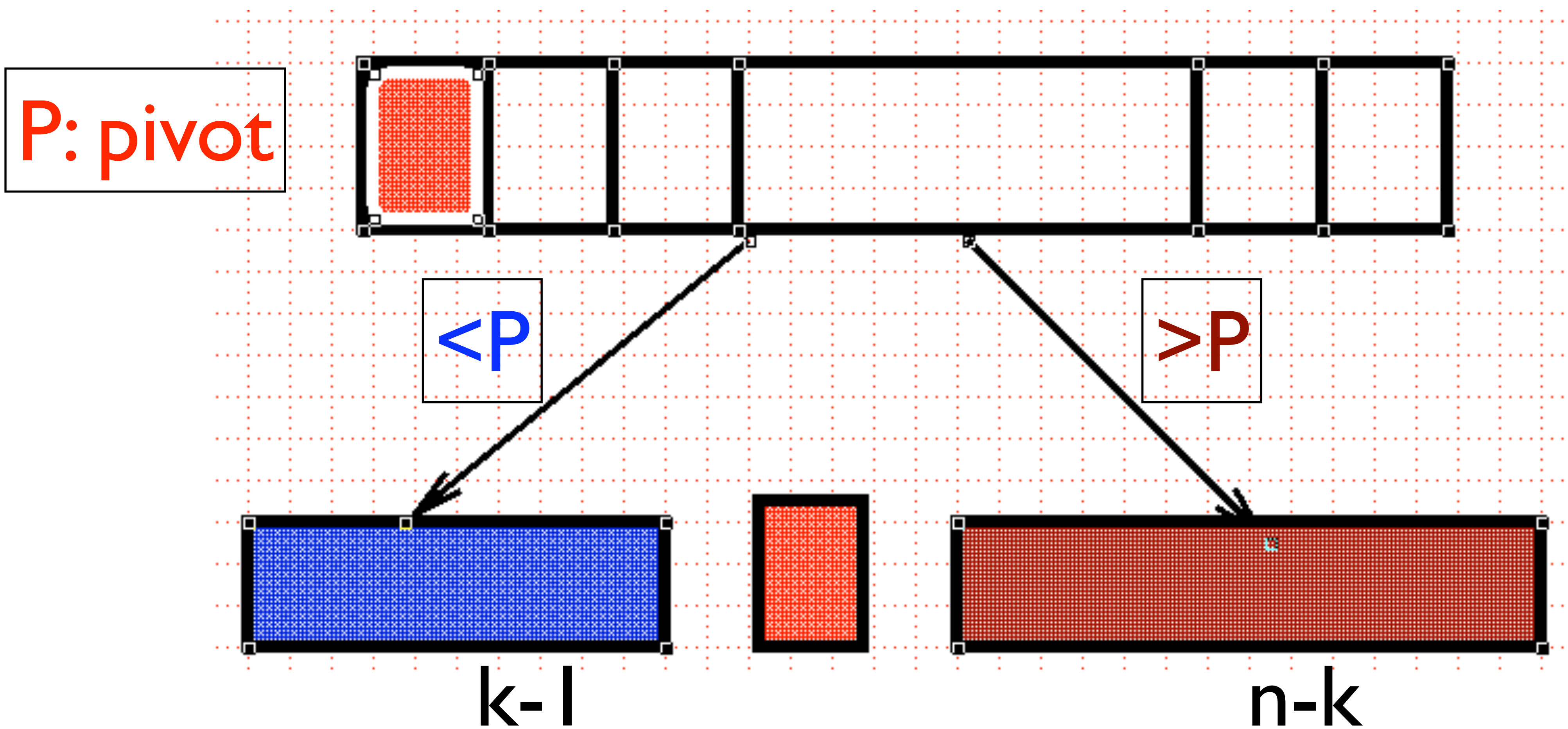
QuickSort (n, A): sorts the array A

Choose a pivot;

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

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

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



Analyses of QuickSort

- *Average-case*: recurrences, then generating functions (GFs). Exchanges; Median-of-3, etc.
- *Variance: multivariate GFs*
- *Distribution*:
  MGFs & moments, Martingales, Contraction

Hoare; Knuth; Sedgewick [1960-1975]
Hennequin, Régnier, Rösler [1989+]
Fill & Janson [2000], Martinez...

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

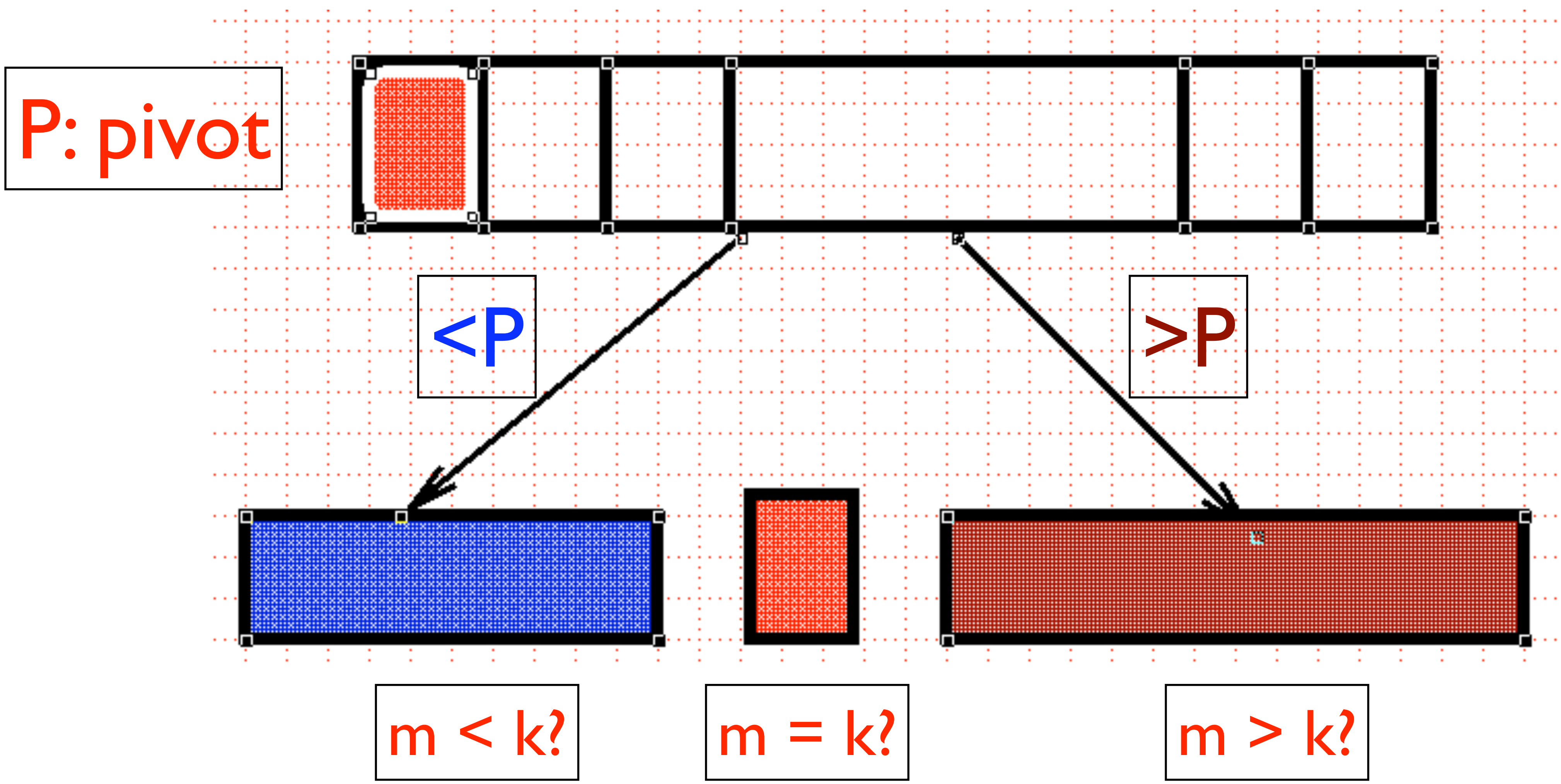
Choose a pivot;

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

If $m = k$ then QuickSelect := pivot

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

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

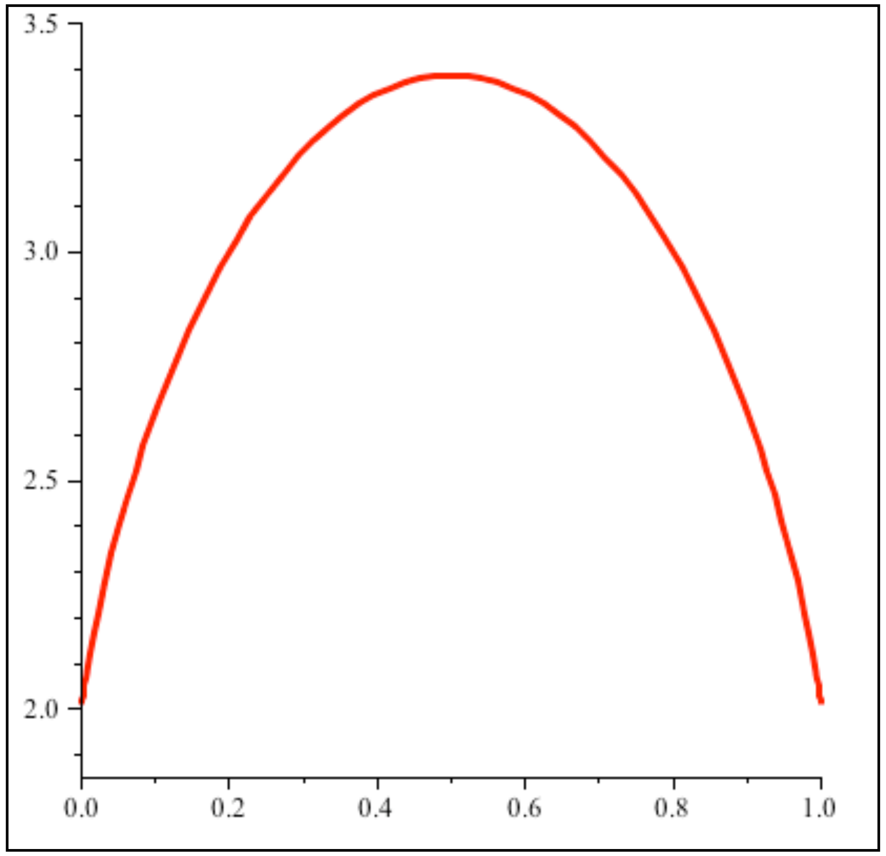


Various brands of QuickSelect:

$\texttt{QuickMin}(n)$	$:= \texttt{QuickSelect}(1, n)$	finds the minimum
$\texttt{QuickMax}(n)$	$:= \texttt{QuickSelect}\ (n, n)$	finds the maximum.
$\texttt{QuickQuant}_\alpha(n)$	$:= \texttt{QuickSelect}\ (\lfloor \alpha n \rfloor, n)$	finds the α–quantile
$\texttt{QuickMed}(n)$	$:= \texttt{QuickSelect}\ (\lfloor n/2 \rfloor, n)$	finds the median
$\texttt{QuickRand}(n)$	$:= \texttt{QuickSelect}\ (m, n)$	for a rank $m \in [1..n]_{\mathcal{R}}$

Average-case analyses

Mean number K_n of key comparisons



$\mathtt{QuickSort}(n)$	$K_n \sim 2n \log n$
$\mathtt{QuickMin}(n)$	$K_n \sim 2n$
$\mathtt{QuickMax}(n)$	$K_n \sim 2n$
$\mathtt{QuickRand}(n)$	$K_n \sim 3n$
$\mathtt{QuickQuant}_\alpha(n)$	$K_n \sim 2\left[1 + H_\alpha\right] n$
$\mathtt{QuickMed}(n)$	$K_n \sim 2(1 + \log 2)n$

$H_\alpha =$ the entropy function $= \alpha|\log \alpha| + (1-\alpha)|\log(1-\alpha)|$

Knuth et al [ca 1970]

Distributional analyses

- Quickselect: e.g., Dickman distribution
  Mahmoud-Modarres-Smythe, Grübel, Rösler, Hwang-Tsai, et al.
  perpetuities: *$1+U_1+U_1U_2+U_1U_2U_3+\ldots$*
  i.i.d. unif. [0,1]
  (fixed rank; fixed quantile)

- Multiple Quickselect, ancestors, &c
  Lent-Mahmoud, Prodinger, et al.

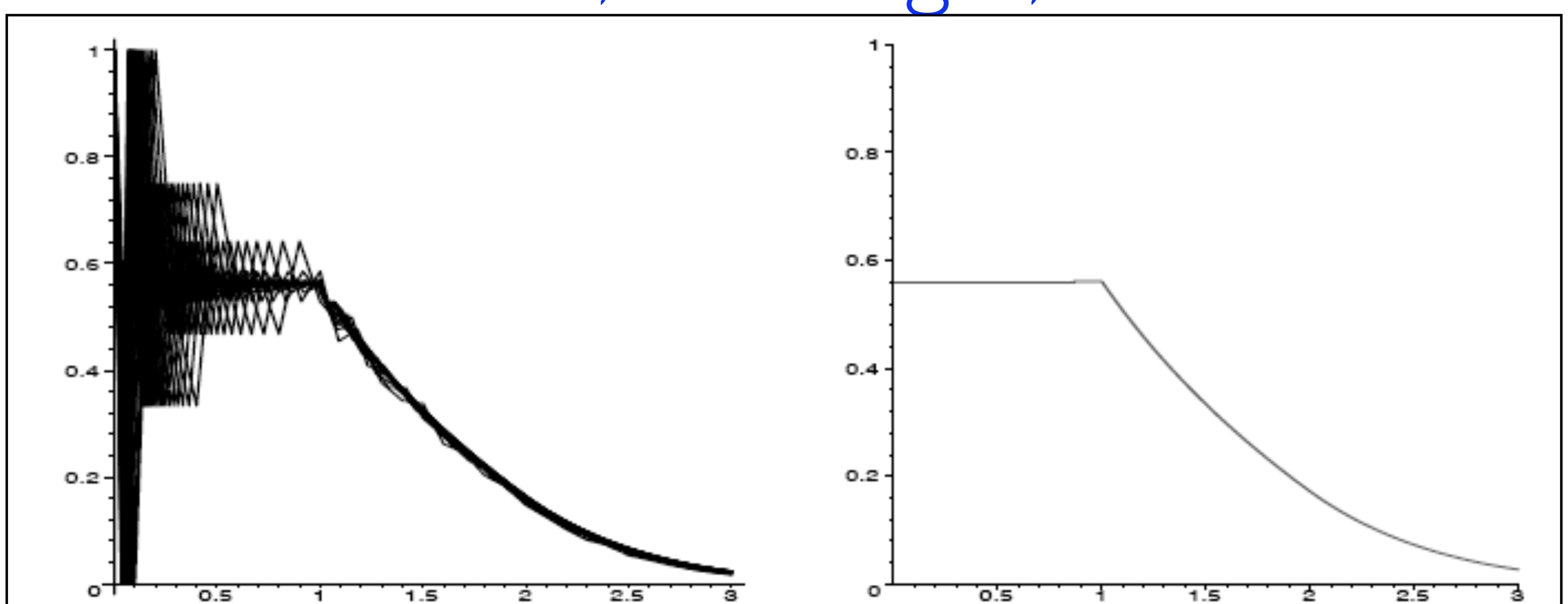


Figure 2 Histograms of $nP(D_n = \lfloor xn \rfloor)$ for n from 10 to 30 and

2. Cost measures

Sedgewick @ AofA-02(?): "actual complexity matters!"

- So far: number of key-comparisons
- But... *keys are often "non-atomic" records!*
- *And...need common information-theoretic basis, to compare with radix methods, hashing, etc.*

Alphabet: Σ

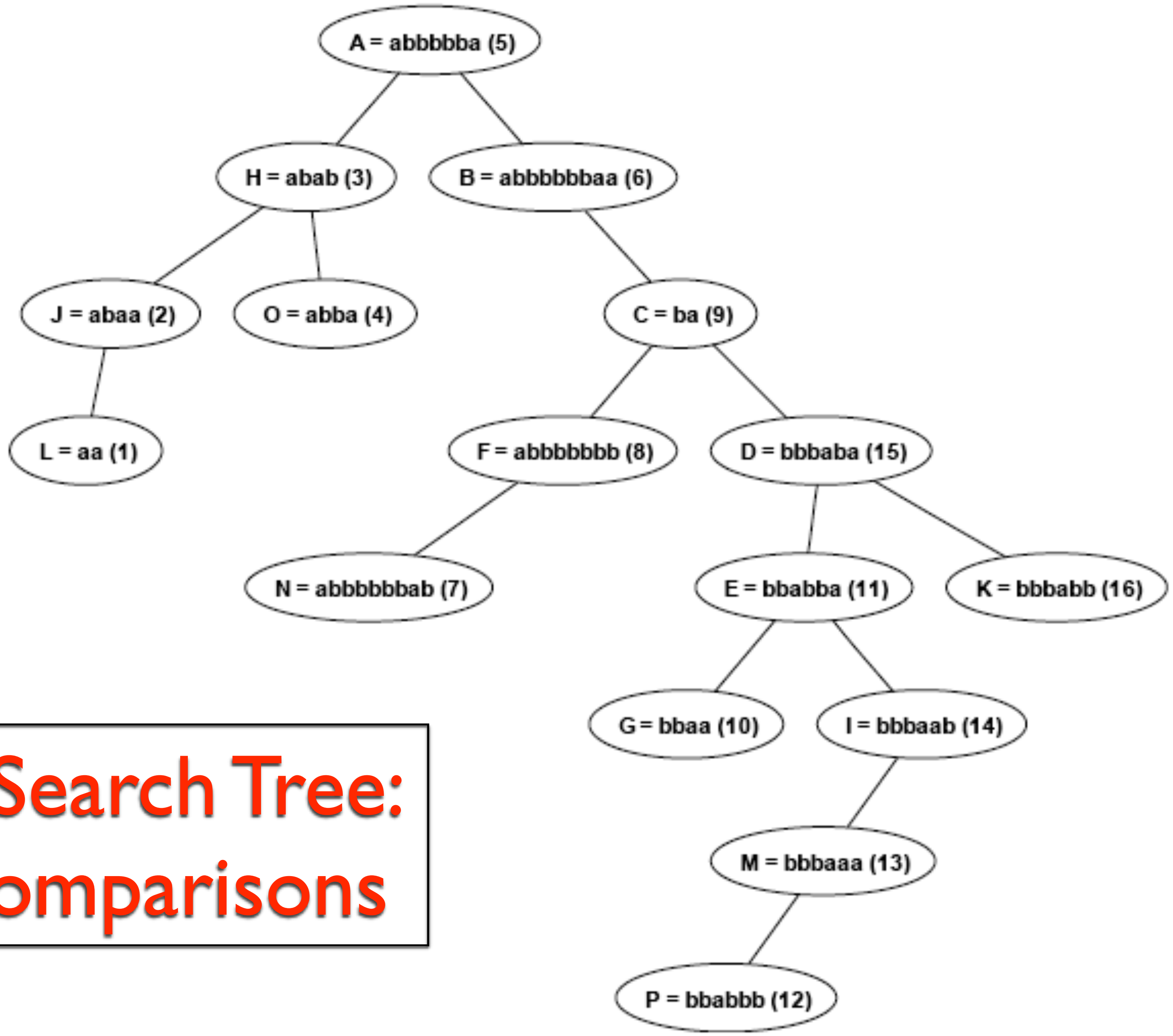
- *Count all symbol comparisons in algorithms:*
- comparing u and v has cost 1+<u>coincidence</u> (u,v).

$$\begin{array}{l} a\ b\ a\ \underline{\mathrm{b}}\ \mathrm{b}\ \mathrm{b}\ldots \\ a\ b\ a\ \underline{\mathrm{a}}\ \mathrm{b}\ \mathrm{a}\ldots \end{array}$$

coincidence=3; #comparisons=4.

(γ) (β)

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 = abbababbbbbb P = bbabbbaaaabb



A Binary Search Tree: symbol comparisons

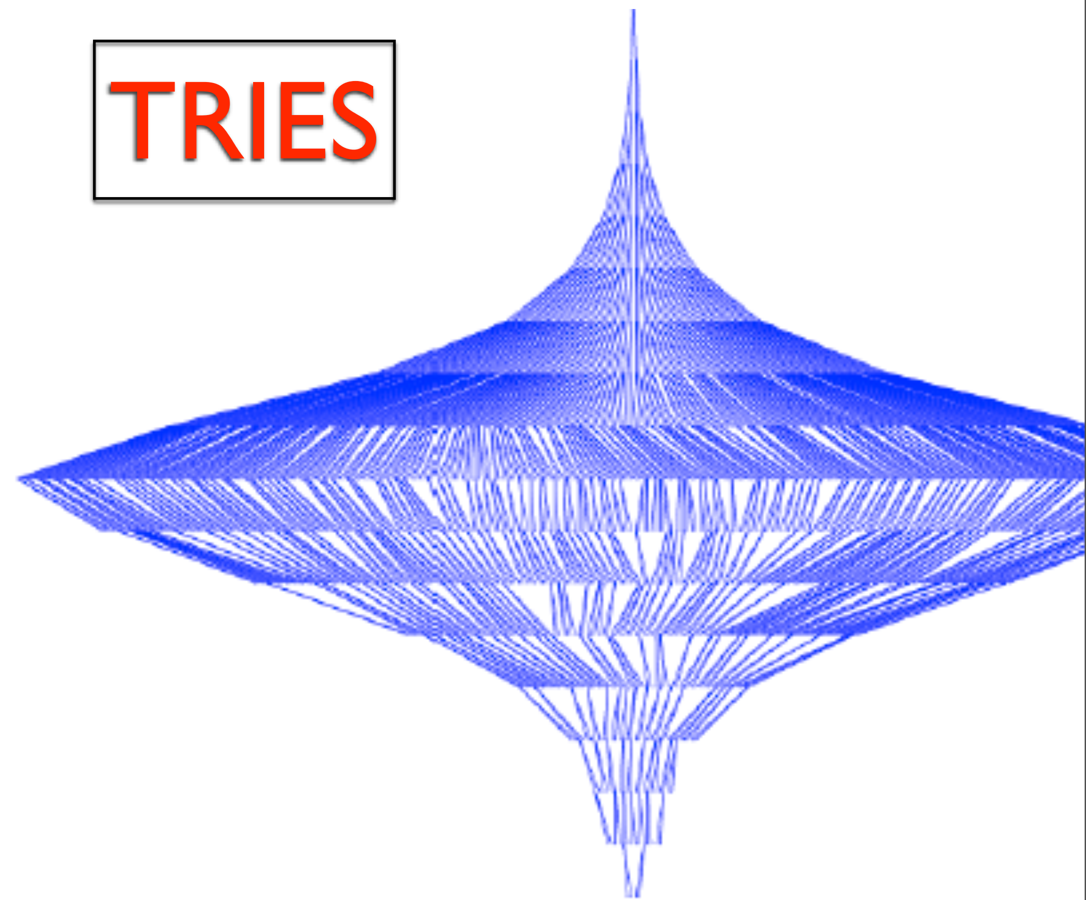
Under a wide range of *classical*
STRING (WORD) MODELS:

It takes O(*n*.log *n*)
<u>symbol comparisons</u> to "distinguish" <u>*n*</u>
elements --- in probability, on average

With high probability,
the <u>common prefix</u> of any two words has length
at most O(log *n*).

Many many people in the audience...

TRIES



- Bernoulli, Markov, etc.
- Devroye's density model
- Vallée's dynamic sources...

$S_n = O(K_n . \log(n))$

Upper bounds

- Quicksort: $O(n.(\log n)^2)$
- Quickselect: $O(n.\log n)$

Symbol comparisons

- QuickSort: [Janson & Fill 2002] *binary source + density model.*

$\sim Cn.\log(n)^2$

- QuickSelect: [Fill-Nakama 2007-9] *binary source for QuickMin/Max & QuickRand*

$\sim C'.n$

CONSTANTS?

(cf also: Panholzer & Prodinger)

3. Sources

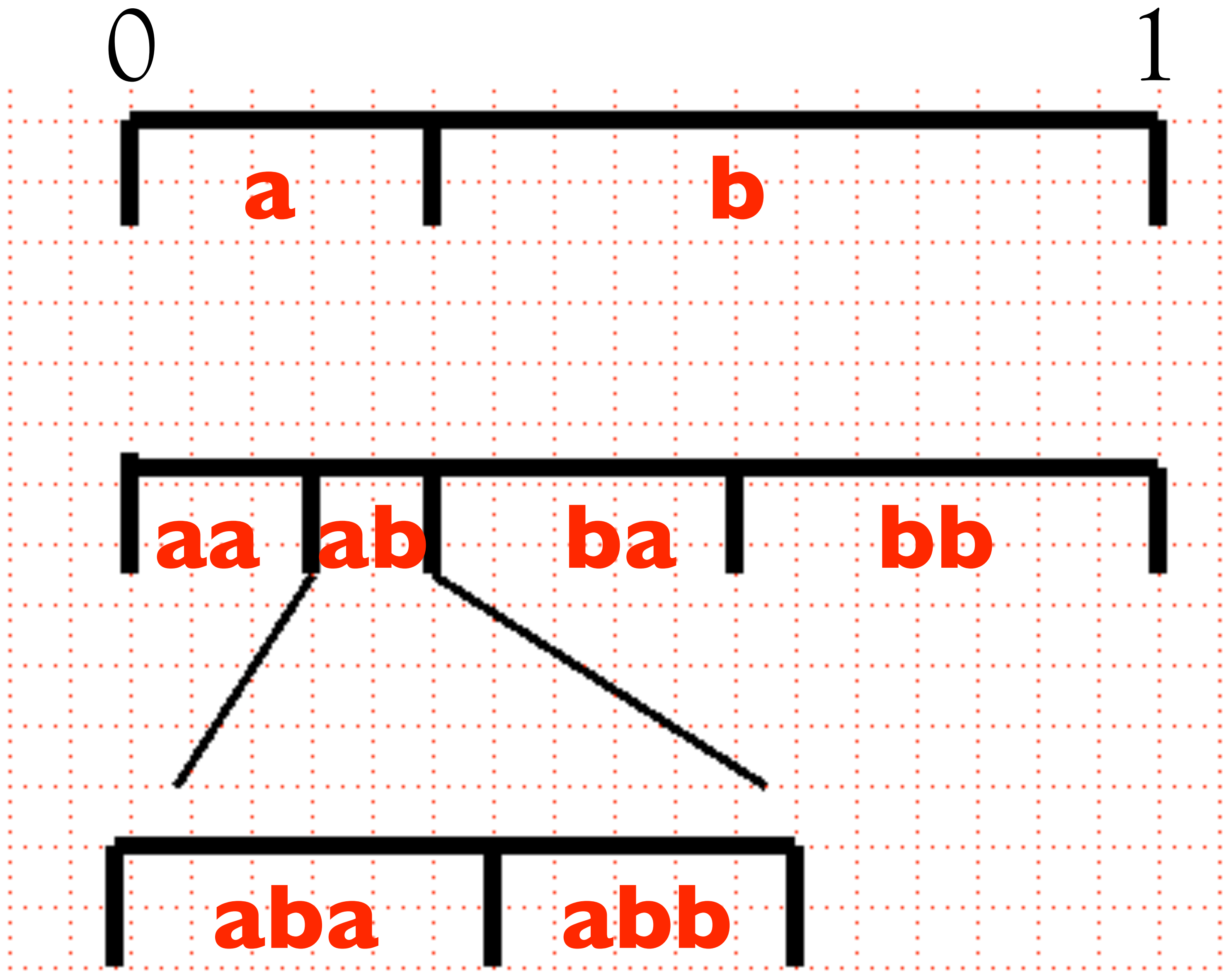
“A source models the way data (symbols) are produced.”

“La Source” by Ingres @ Musée d’Orsay

Axioms for SOURCES

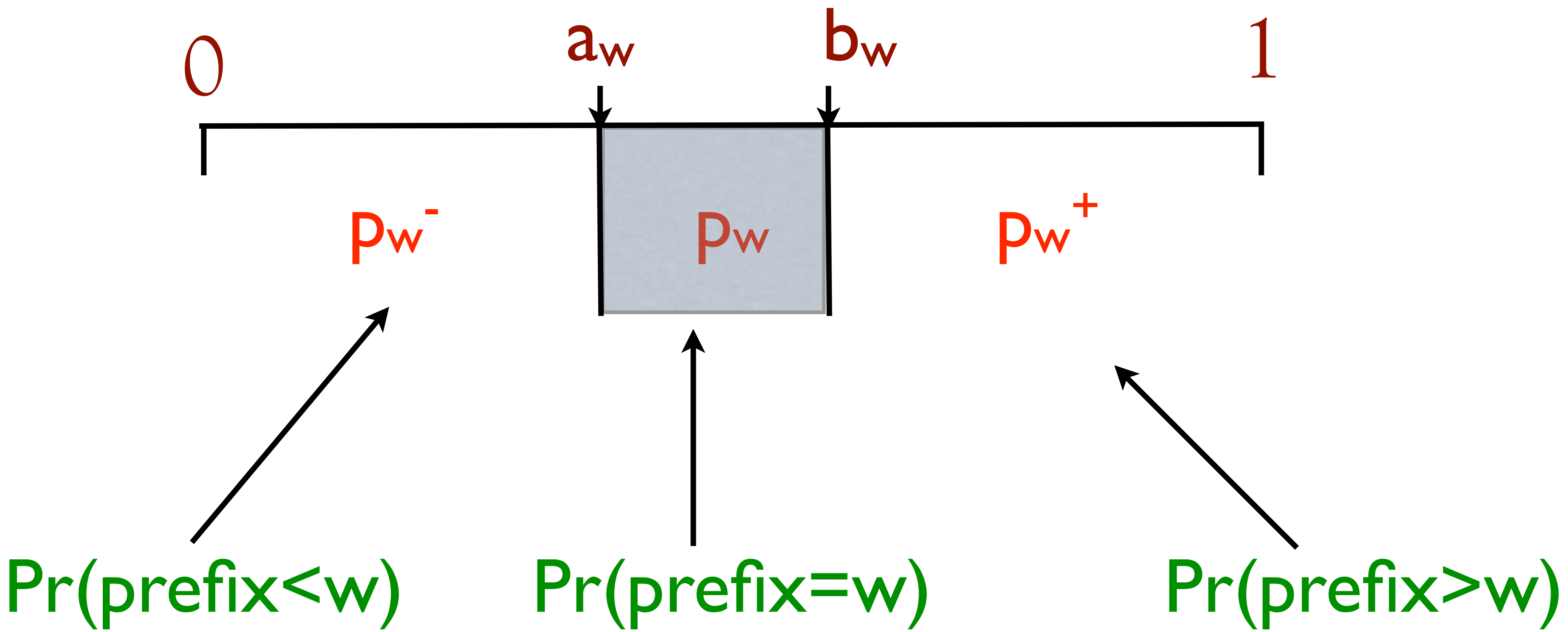
- Totally ordered alphabet (usually finite) Σ
- Fundamental probabilities (p_w) :=
  the probability of starting with w
- $p_w \to 0$ as $|w| \to \infty$
- *Keys are invariably i.i.d.*

[Later] + "regularity" conditions:
tameness



Property: *The Source is parameterized by [0,1]: to an infinite word* <u>*w*</u>*, there corresponds* <u>*α*</u> *such that* $M(\alpha)=w$.

Notations:



Fundamental constants of QuickStuffs will be all expressed in terms of fundamental probabilities

- Standard binary source (uniform: 1/2,1/2); Bernoulli sources such as 1/2, 1/6,1/3.

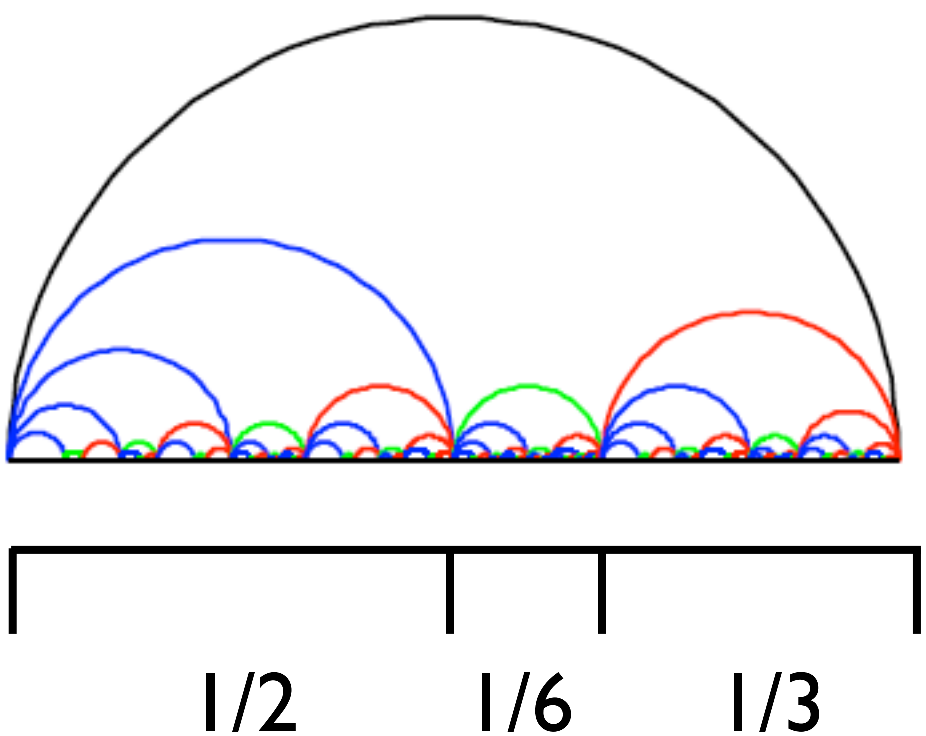
- *Density models:* Standard binary source with density f(x) or c.d.f F(x).

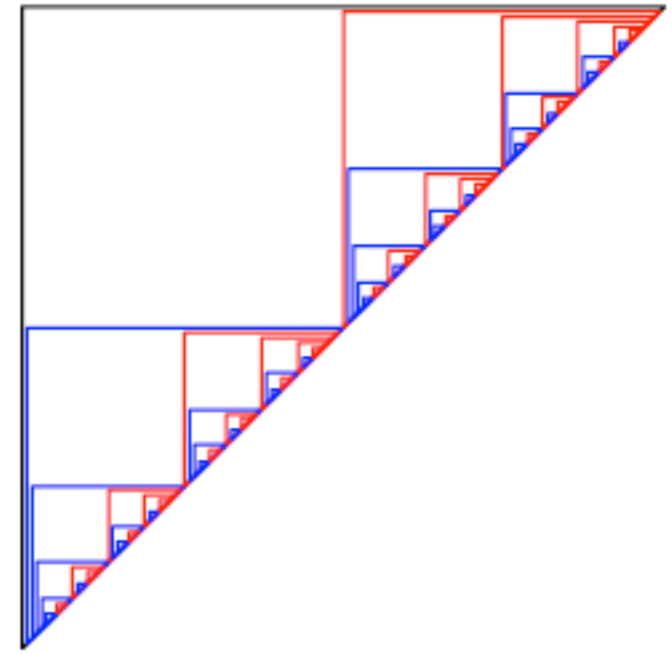
- Markov

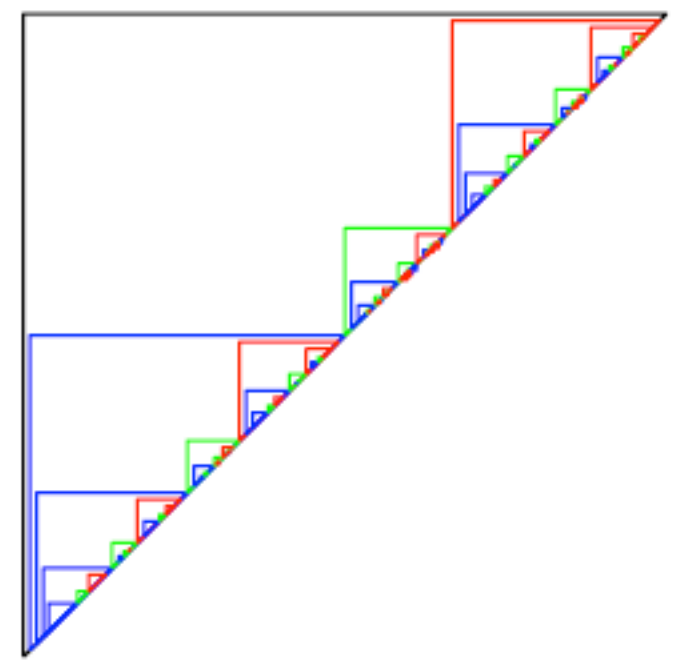
- *Dynamical sources*

[Devroye 1986] [Vallée 2001; Clément-Fl-Vallée 2001]

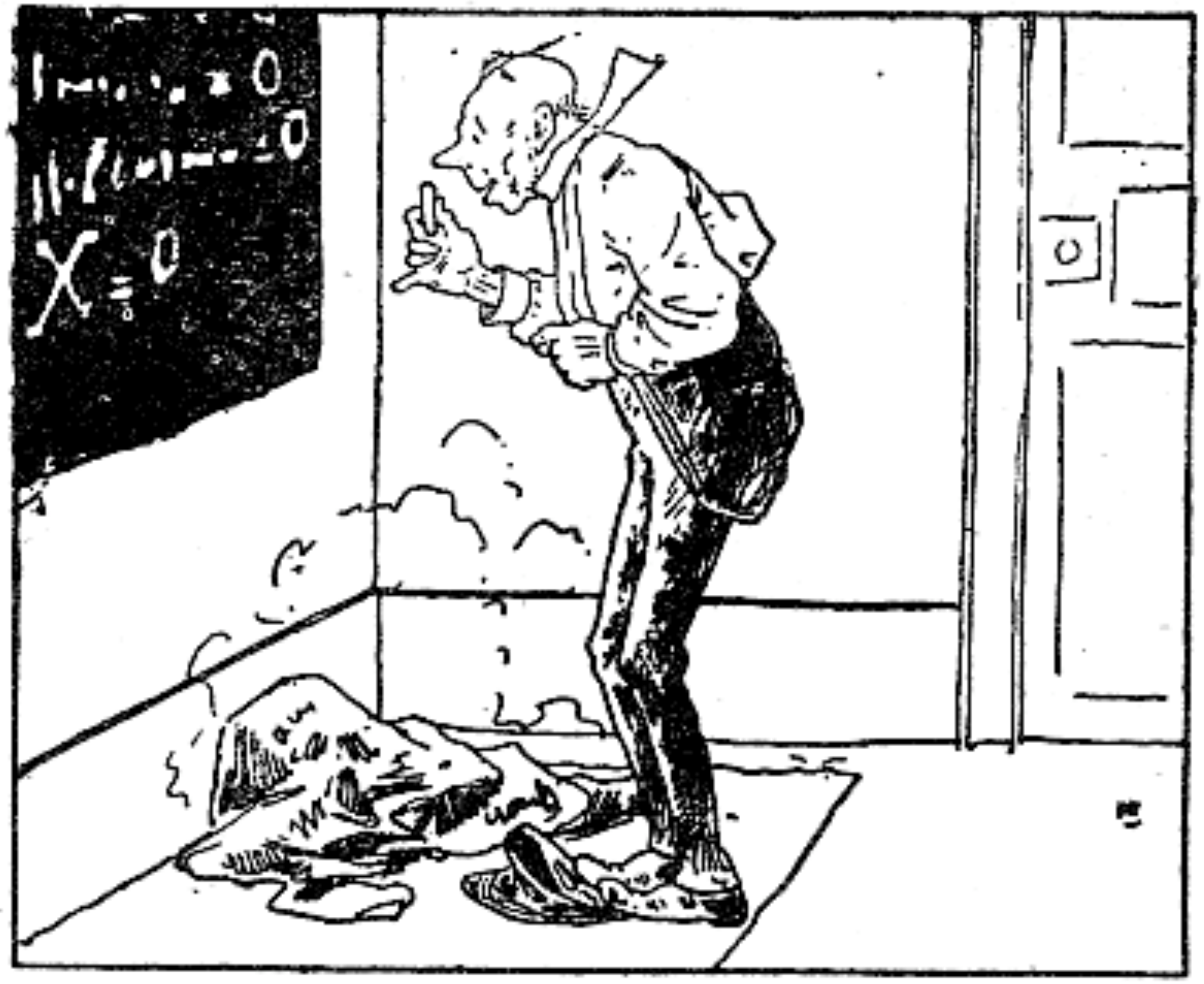
Fundamental intervals & triangles







4. Results



(Le Savant Cosinus)

Average-case

Theorem 1 *For any tamed source, the mean number S_n of symbol comparisons used by* `QuickSort`(n) *satisfies*

$$\boxed{S_n \sim \frac{1}{h_{\mathcal{S}}}\, n \log^2 n.}$$

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

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

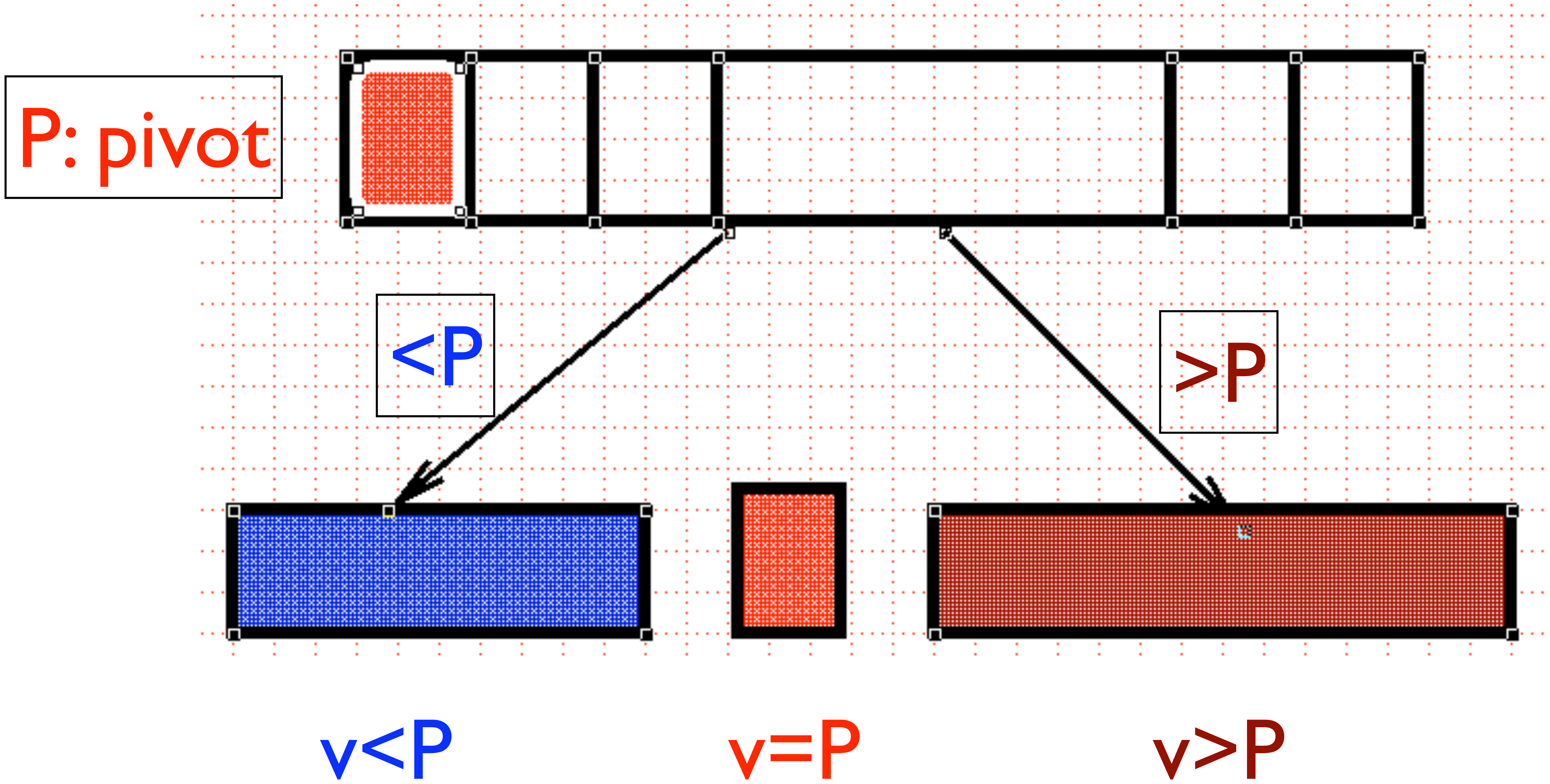
where p_w is the probability that a word begins with prefix w.

Theorem 2 *For any weakly tamed source, the mean number of symbol comparisons used by* `QuickQuant`$_\alpha(n)$ *satisfies* $\boxed{q_n^{(\alpha)} \sim \rho_{\mathcal{S}}(\alpha)\, n}$

➔ ➔ QuickMin, QuickRand

QuickVal ☞ ☞

QUICKVAL(α): is dual to QuickSelect



- QuickVal(n,$\boldsymbol{\alpha}$) := *rank of element whose parameter [corresponding to value* **v***] is* $\boldsymbol{\alpha}$.

- QuickVal(n,α) behaves “almost” as QuickSelect($n\alpha$).

Distribution

Theorem: *Assuming a suitable tameness condition, there exists a limiting distribution of the cost* S_n/n *of* QuickQuant(α), *which can be described explicitly*