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;

\((k, A_-, A_+) := \text{Partition}(A);\)

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;
\((k, A_-, A_+) := \text{Partition}(A)\);
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:

- \text{QuickMin}(n) := \text{QuickSelect}(1,n) \quad \text{finds the minimum}
- \text{QuickMax}(n) := \text{QuickSelect}(n,n) \quad \text{finds the maximum.}
- \text{QuickQuant}_\alpha(n) := \text{QuickSelect} \left( \lfloor \alpha n \rfloor, n \right) \quad \text{finds the } \alpha \text{-quantile}
- \text{QuickMed}(n) := \text{QuickSelect} \left( \lfloor n/2 \rfloor, n \right) \quad \text{finds the median}
- \text{QuickRand}(n) := \text{QuickSelect}(m,n) \quad \text{for a rank } m \in [1..n] \cap
Average-case analyses

Mean number $K_n$ of key comparisons

<table>
<thead>
<tr>
<th>Function</th>
<th>$K_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>QuickSort$(n)$</td>
<td>$K_n \sim 2n \log n$</td>
</tr>
<tr>
<td>QuickMin$(n)$</td>
<td>$K_n \sim 2n$</td>
</tr>
<tr>
<td>QuickMax$(n)$</td>
<td>$K_n \sim 2n$</td>
</tr>
<tr>
<td>QuickRand$(n)$</td>
<td>$K_n \sim 3n$</td>
</tr>
<tr>
<td>QuickQuant$_\alpha(n)$</td>
<td>$K_n \sim 2[1 + H_\alpha] n$</td>
</tr>
<tr>
<td>QuickMed$(n)$</td>
<td>$K_n \sim 2(1 + \log 2)n$</td>
</tr>
</tbody>
</table>

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

Knuth et al [ca 1970]
Distributional analyses

- Quickselect: e.g., Dickman distribution
  Mahmoud-Modarres-Smythe, Grübel, Rösler, Hwang-Tsai, et al.
  \[ \text{perpetuities: } 1 + U_1 + U_1 U_2 + U_1 U_2 U_3 + \ldots \]
  \[ \text{i.i.d. unif. } [0, 1] \]
  (fixed rank; fixed quantile)

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

---

Figure 2: Histograms of \( nP(D_n = [\eta n]) \) for \( n \) from 10 to 30 and 6, 64, 6 and the Dickman distribution \( \xi_x \).
2. Cost measures
• 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: $\Sigma$

- **Count all symbol comparisons in algorithms:**

- comparing $u$ and $v$ has cost $1 + \text{coincidence}(u,v)$.

\[
\begin{array}{cccccccc}
 a & b & a & b & b & b & \ldots \\
 a & b & a & a & b & a & \ldots \\
\hline
\end{array}
\]

\text{coincidence}=3; \quad \#\text{comparisons}=4.

(\gamma) \quad (\beta)
A Binary Search Tree: symbol comparisons

A = abbbbbaaabab  B = abbbbbbaabaa  C = baabbbabbbba  D = bbbababbbbaab  E = bbabbaababbb
F = abbbbbbbbbabb  G = bbaabbbababa  H = ababbbabbbbab  I = bbbaabbbbbbbb  J = ababbbbabbaab
K = bbabbbbbbbaa  L = aabbaababaab  M = bbbababbb  N = ababbbbbababaa  O = ababababbb  P = bbabbbbaab
Under a wide range of *classical* STRING (WORD) MODELS:

It takes $O(n \cdot \log n)$ symbol comparisons to “distinguish” $n$ elements --- in probability, on average.

With high probability, the common prefix of any two words has length at most $O(\log n)$.

**Many many** people in the audience...
• Bernoulli, Markov, etc.
• Devroye’s density model
• Vallée’s dynamic sources...

\[ S_n = O(K_n \log(n)) \]

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

Upper bounds
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

\[ \sim C' \cdot n \]

CONSTANTS?

(cf also: Panholzer & Prodinger)
3. Sources

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

- Totally ordered alphabet (usually finite) $\Sigma$
- 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 $w$, there corresponds $\alpha$ such that $M(\alpha) = w$. 
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-Fi-Vallée 2001]
Fundamental intervals & triangles

\[
\frac{1}{2} \quad \frac{1}{6} \quad \frac{1}{3}
\]
4. Results

(Le Savant Cosinus)
**Theorem 1.** For any tamed source, the mean number $S_n$ of symbol comparisons used by $\text{QuickSort}(n)$ satisfies

$$S_n \sim \frac{1}{h_S} n \log^2 n,$$

and involves the entropy $h_S$ of the source $S$, defined as

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

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 $\text{QuickQuant}_\alpha(n)$ satisfies $q_n^{(\alpha)} \sim \rho S(\alpha) n$.

**QuickMin, QuickRand**

**QuickVal**
**QUICKVAL(α):** is dual to QuickSelect

- \( \text{QuickVal}(n, \alpha) := \text{rank of element whose parameter corresponding to value } v \) is \( \alpha \).
- \( \text{QuickVal}(n, \alpha) \) behaves “almost” as \( \text{QuickSelect}(n\alpha) \).
Theorem: Assuming a suitable tameness condition, there exists a limiting distribution of the cost $S_n/n$ of QuickQuant($\alpha$), which can be described explicitly.