Comparison Problems

**input**

set X of N distinct keys

total ordering < over X

**Problems**

1. for each key x $\in X$
   
   \[ \text{rank}(x, X) = |\{ x' \in X | x' < x \}| + 1 \]

2. for each index i $\in \{1, ..., N\}$
   
   \[ \text{select}(i, X) = \text{the key } x \in X \text{ where } i = \text{rank}(x, X) \]

3. \[ \text{sort}(X) = (x_1, x_2, ..., x_n) \]
   
   where $x_i = \text{select}(i, X)$

---

**ALG 2.1**

**Randomized Algorithms for Selection and Sorting:**

(a) Randomized Sampling
(b) Selection by Randomized Sampling
(c) Sorting by Random Splitting: Quicksort and Multisample Sorts

**Main Reading Selections:**
- CLR, Chapters 8, 10

**Auxillary Reading Selections:**
- AHU-Design, Sections 3.5-3.7
- BB, Sections 4.5, 4.6
- AHU-Data, Section 8.3
- Handout: "Derivation of Randomized Algorithms"
**Randomized Comparison Tree Model**

1. **Comparison nodes**
   - $X_i < X_j$
   - Yes \(\rightarrow\) \(\frac{1}{2}\)\(\rightarrow\) Yes
   - No \(\rightarrow\) \(\frac{1}{2}\)\(\rightarrow\) No

2. **Random Choice Nodes**
   - RANDOM
   - prob $\frac{1}{2}$

**Algorithm samplerank \(_s(x,X)\)**

```
begin
Let S be a random sample of X-{x} of size s
output 1 + \frac{N}{s} [rank (x,S) - 1]
end
```
Lemma 1

The expected value of samplerank \( s(x,X) \) is rank \( (x,X) \)

Proof

Let \( k = \text{rank}(x,X) \)

For a random \( y \in X \),

\[
\text{Prob}(y < x) = \frac{k - 1}{N}
\]

Hence \( E(\text{rank}(x,S)) = s \cdot \frac{k - 1}{N} + 1 \)

Solving for \( k \), we get

\[
\text{rank}(x,X) = k = 1 + \frac{N}{s} \cdot E[\text{rank}(x,S) - 1]
\]

\[
= E(\text{sample rank}(x,X))
\]
More Precise Bounds on Randomized Sampling

Let $S$ be a random sampling of $X$

Let $r_i = \text{rank}(\text{select } (i, S), X)$

**Lemma 2**

$\text{Prob}\left( |r_i - \frac{iN}{s+1}| > \frac{cN}{\sqrt{s}} \sqrt{\log N} \right) < N^{-\alpha}$

**proof**

We can bound $r_i$ by a Beta distribution, implying

$\text{mean } (r_i) = \frac{iN}{s+1}$

$\text{Var } (r_i) \leq \frac{i(s-i+1)}{(s+1)^2(s+2)} N^2$

Weak bounds follow from Chebychev inequality

The Tighter bounds follow from Chernoff Bounds
Subdivision by Random Sampling

Let $S$ be a random sample of $X$ of size $s$

Let $k_1, k_2, ..., k_s$ be the elements of $S$ in sorted order

These elements subdivide $X$ into $s+1$ subsets

$x_1 = \{x \in X \mid x \leq k_1\}$

$x_2 = \{x \in X \mid k_1 < x \leq k_2\}$

$x_3 = \{x \in X \mid k_2 < x \leq k_3\}$

$\vdots$

$x_{s+1} = \{x \in X \mid x > k\}$

How even are these subdivisions?

Lemma 3

If random sample $S$ in $X$ is of size $s$ and $X$ is of size $N$, then $S$ divides $X$ into subsets each of size $\leq \alpha \frac{(N-1)}{s} \ln(N)$ with prob $\geq 1 - N^{-\alpha}$
proof

The number of \((s+1)\) partitions of \(X\) is
\[
\frac{\binom{N-1}{s}}{s!} \sim \frac{(N - 1)^s}{s!}
\]

The number of partitions of \(X\) with one
block of size \(\geq v\) is
\[
\frac{\binom{N - v - 1}{s}}{s!} \sim \frac{(N - v - 1)^s}{s!}
\]

So the probability of a random \((s + 1)\) partition
having a block size \(\geq v\) is
\[
\frac{\binom{N - v - 1}{s}}{s!} \sim \left(\frac{N - v - 1}{N - 1}\right)^s \leq \left(1 - \frac{1}{Y}\right)^Y \leq e^{-sY}
\]

\[= e^{-sY} \leq N^{-\alpha} \text{ if } v = \alpha \frac{(N-1)}{s} \ln N \]

since \(\left(1 - \frac{1}{Y}\right)^Y < e^{-1}\)

Randomized Algorithms for Selection

"canonical selection algorithm"

Algorithm

can select \((i,X)\)
input set \(X\) of \(N\) keys
index \(i \in \{1,...,N\}\)

[0] if \(N=1\) then output \(X\)

[1] select a bracket \(B\) of \(X\), so that
select \((i,X) \in B\) with high prob.

[2] Let \(i_1\) be the number of keys
less than any element of \(B\)

[3] output can select \((i-i_1, B)\)

Note: \(B\) found by random sampling
(also must cover cases of low prob
to always get correct output)
Hoar's Selection Algorithm

Algorithm \( \text{Hselect}(i, X) \) where \( 1 \leq i \leq N \)

\[
\begin{align*}
\text{begin} & \quad \text{if} \quad X = \{x\} \quad \text{then output} \quad x \quad \text{else} \\
& \quad \text{choose a \ random splitter \ } k \in X \\
& \quad \text{let } B = \{x \in X \mid x < k\} \\
& \quad \text{if} \quad |B| \geq i \quad \text{then output} \quad \text{Hselect}(i, B) \\
& \quad \text{else output} \quad \text{Hselect}(i - |B|, X - B) \\
\text{end}
\end{align*}
\]

Sequential time bound \( T(i, N) \) has mean

\[
T(i, N) = N + \frac{1}{N} \left[ \sum_{j=1}^{i} T(i-j, N-j) + \sum_{j=i+1}^{N} T(i, j) \right]
\]

\[
= 2N + \min(i, N-i) + o(N)
\]

random splitter \( k \in X \) \( \text{Hselect}(i, X) \) has two cases

**Case** \( |B| < i \)

\[
B \quad k \quad X-B
\]

\[
i \quad i - |B|
\]

**Case** \( |B| \geq i \)

\[
B \quad k \quad X-B
\]

\[
i
\]

Inefficient: each recursive call requires \( N \) comparisons, but only reduces problem size by average \( \frac{1}{2} N \)
Improved Randomized Selection
by Floyd and Rivest

Algorithm
FRselect(i, X)

begin
  if X = {x} then output x else
    Choose k₁, k₂ ∈ X such that k₁ < k₂
    let r₁ = rank(k₁, X), r₂ = rank(k₂, X)
    if r₁ > i then FRselect(i, {x ∈ X | x < k₁})
    else if r₂ > i then FRselect(i - r₁, {x ∈ X | k₁ ≤ x ≤ k₂})
    else FRselect(i - r₂, {x ∈ X | x > k₂})
end

problem:
We must choose k₁, k₂ so that with high likelihood, k₁ ≤ select(i, X) ≤ k₂

Choose random sample S ⊆ X size s

Define:

\[
k₁ = \text{select}
\left(i \frac{(s+1)}{(N+1)} - \delta, S\right)
\]
\[
k₂ = \text{select}
\left(i \frac{(s+1)}{(N+1)} + \delta, S\right)
\]

where \( \delta = \left\lceil \sqrt{d \alpha s \log N} \right\rceil \), d=constant

Lemma 2 implies:

\[
\Pr (r₁ > i) < N^{-α}
\]
and
\[
\Pr (r₂ < i) < N^{-α}
\]
where \( r₁ = \text{rank} (k₁, X) \)
\( r₂ = \text{rank} (k₂, X) \)
Expected Time Bound

\[ T(i, N) \leq N + 2T(-, s) \]
\[ + \text{Prob}(r_1 > i) \cdot T(i, r_1) \]
\[ + \text{Prob}(i > r_2) \cdot T(i - r_1, N - r_2) \]
\[ + \text{Prob}(r_1 \leq i \leq r_2) \cdot T(i - r_1, r_2 - r_1) \]

\[ \leq N + 2T(-, s) + 2N^{-\alpha} \cdot N + T\left(i, 2 \left[ \frac{N+1}{s+1} \right] \right) \]

\[ \leq N + \min(i, N-i) + o(N) \]

if we set \( \delta < \frac{3}{\alpha} \) and \( s = N^3 \log N \)

= \( o(N) \)

**note**

with prob \( \geq 1 - 2N^{-\alpha} \) each

recursive call costs only \( O(s) = o(N) \)

rather than \( N \) in previous algorithm
Randomized Sorting Algorithms

"canonical sorting algorithm"

Algorithm
cansort(X)
begin
if x = {x} then output X else
choose a random sample S of X of size s
Sort S
S subdivides X into s+1 subsets
X_1, X_2, ..., X_{s+1}
output cansort(X_1) cansort(X_2) ... cansort(X_{s+1})
end

Problem:
must subdivide X into subsets of nearly equal size to minimize number of comparisons

Solution: random sampling!

Hoar's Randomized Sorting Algorithm

uses sample size s=1

Algorithm quicksort(X)
begin
if |X|=1 then output X else
choose a random splitter k \in X
output quicksort({x \in X | x<k}) \cdot (k) \cdot quicksort({x \in X | x>k})
end

Expected Time Cost

\[ \bar{T}(N) \leq N - 1 + \frac{1}{N} \sum_{i=1}^{N} (\bar{T}(i-1) + \bar{T}(N-i)) \]

\[ \leq 2 \cdot N \log N \]

inefficient:
need to divide problem size by \( \frac{1}{2} \) with high likelihood!
Better choice splitter is

\[ k = \text{sample select}_s \left( \left\lfloor \frac{N}{2} \right\rfloor, N \right) \]

**Algorithm samplesorts (X)**

```
begin
  if \(|X|=1\) then output \(X\)
  choose a random subset \(S\) of \(X\) size \(s=N/\log N\)
  \(k \leftarrow \text{select}(\left\lfloor s/2 \right\rfloor, S)\) cost \( \Theta(N) \)
  output samplesorts_{\{x \in X \mid x<k\}}(k) \cdot \text{samplesorts}_{\{x \in X \mid x>k\}}(S)
end
```

By Lemma 2, \(\text{rank}(k,X)\) is very nearly the mean:

\[
\Pr(|\text{rank}(k,X) - \frac{N}{2}| > \sqrt{d \log N}) < N^{-\alpha}
\]

Expected Time Bounds

\[
\bar{T}(N) \leq 2\bar{T}(N_1) + N^{-\alpha} \quad \bar{T}(N) \cdot N + o(N) + N - 1
\]

\(\log (N!)\) is optimal for comparison trees!
Open Problems in Selection and Sorting

(1) *Improve* Randomized Algorithms
to *exactly match lower bounds* on number of comparisons

(2) Can we *de* randomize these algorithms -
    i.e., give *deterministic algorithms* with the same bounds?