Odds + Ends

Sorting, Selecting, Searching

We can sort in $O(n \log n)$ time.

Quicksort (average)

Mergesort (worst-case)

Is this bound tight?
Decision Tree Model

Input: a permutation of \( n \) numbers

At each node, ask any yes-no question about the permutation (comparisons a special case)

Each permutation reaches a different leaf

Depth = # questions ≤ time

Too broad: no accounting for deciding what questions to ask; program need not be of fixed size.

Too narrow: only binary decisions
There are $n!$ permutations on $n$ items.

$\lceil \log_2 (n!) \rceil = \lceil \log_2 (n!) \rceil$ is a lower bound on the worst-case depth.

Stirling's approximation for $n!$

$$n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$$

$$\Rightarrow \left\lceil \log_2 n! \right\rceil \geq n \log_2 n - n/\ln 2$$

$$\Rightarrow \text{sorting takes } n \log_2 n - n/\ln 2$$

comparisons (or binary decisions) \hspace{1cm} \text{(worst-case)}

\hspace{1cm} \text{Information-Theoretic Lower Bound}
Average case?

Lower bound: sum of depths of leaves / n!

sum of depths of leaves = external path length

minimized for N external nodes

at \((q+1)N - 2^q\) where \(q = \lfloor \lg N \rfloor\)

(complete binary tree: all leaves on at most 2 levels)

\(EPL = N \lg N + 1 + \Theta - 2^\Theta\)

where \(\Theta = \lg N + \Theta\)

\(1 + \Theta - 2^\Theta \geq 0\)

\(\leq 0.0861\)

For sorting, ave length is \(\geq EPL(n!) / n!\)

\(\geq \lg n! = n\lg n - n / \ln 2\)
Average case analysis of quicksort

Compare each item to splitter 
(n-1 comparisons)

Recursively sort smaller set, larger set

\[ A_n = n - 1 + \frac{1}{n} \sum_{k=0}^{n-1} (A_k + A_{n-k-1}) \]

\[ A_0 = A_1 = 0 \]

\[ A_n / n + 1 \] is also the average depth of an unbalanced binary tree constructed by random insertions

\[ \text{random} \quad \text{random} \]
Observe: items $i$ and $j$ are compared iff one of them is chosen as a splitter before anything in between. The chance of this is $2/(j-i+1)$.

Expected # comparisons:

$$\sum_{i=1}^{n-1} \sum_{j>i}^{n} \frac{2}{(j-i+1)} = \sum_{i=1}^{n-1} \frac{n-i+1}{2}$$

$$= \sum_{k=2}^{n} \frac{1}{k}$$

$$\leq 2 n \ln(n) - O(n) \approx 1.3863 n \ln n$$

$$H = \frac{1}{\sum_{k=1}^{n} \ln k} \leq \ln n + 1$$
Worst-case analysis of merge sort

Divide into two halves

Sort halves

Merge

\[ W(n) = W(\lfloor n/2 \rfloor) + W(\lceil n/2 \rceil) + n - 1 = \sum_{k=1}^{n} \lceil \log k \rceil \]

\[ W(n) \leq n \lceil \log n \rceil = 2^\lceil \log n \rceil + 1 = n \log n - O(n) \]
Selection in $O(n)$ time

Find the $k^{th}$ out of $n$

Quick select: like quick sort, but only recur in the half containing the desired item

if small half contains $i \geq k$, look for $k^{th}$ in small half

if small half contains $k-1$, splitter is $k^{th}$

else look for $k-(i+1)^{st}$ in large half

$O(n)$ on average (analysis like quicksort)
Worst-case $O(n)$: double recursion

Divide into $\lceil n/5 \rceil$ sets of 5 (or less)

Find medians of 5

Find medians of medians

Use $m$ of $m$ as a splitter

Recur in appropriate half

$S(n) = O(n) + S(n/5) + S(3n/10)$

$\frac{3n}{10}$ discarded

linear since $\frac{1}{5} + \frac{3}{10} = \frac{9}{10} < 1$
Optimum Search Trees

Lower bound on weighted (external) path length in a binary tree (entropy bound)

Positive weights $p_i$, $P = \sum p_i$

$$\text{WEPL}' \geq \sum_{i=1}^{n} p_i \log(P/p_i)$$

$\text{WEPL}'$ counts edges, not nodes

($P$ to get node bound)

Proof by induction
\[ WEPL' \geq P + \sum_{i=1}^{k} p_i \log\left(\frac{p_i}{p_i'}\right) + \sum_{i=k+1}^{n} p_i \log\left(\frac{P-p_i'}{p_i}\right) \]

\[ \geq \sum_{i=1}^{n} p_i \log\left(\frac{p_i}{p_i'}\right) + f\left(p'\right) \]

for some \( k \), where

\[ f\left(p'\right) = p' \log p' + \left(p - p'\right) \log\left(\frac{p - p'}{p}\right) - P \log P \]

\( f\left(p'\right) \) is non-negative, taking its min val 0 at \( p' = P/2 \).
This bound is achievable to within $P$
(non-alphabetic: Huffman codes)
or to within $2P$
(alphabetic: Knuth-(internal) or
Hu-Tucker(external))

What about internal path length?

\[
\text{WIPL} \geq \sum_{i=1}^{n} p_i \cdot \lg(P/p_i) - \lg \left( \sum_{i=1}^{n} p_i \cdot \lg(P/p_i) \right) - O(1)
\]

if $P = 1$