Design and Analysis of Algorithms I

Linear-Time Selection

An $\Omega(n \log n)$ Sorting Lower Bound

A Sorting Lower Bound

Theorem: every "comparison-based" sorting algorithm has worst-case running time $\Omega(n \log n)$.

[assume deterministic, but lower bound extends to randomized]

Comparison-Based Sort: accesses input array elements only via comparisons.

≈ "general-purpose sorting method"

Examples: MergeSort, QuickSort, HeapSort

Non-Examples: BucketSort, CountingSort, RadixSort

good for data from distributions

good for small integers

good for medium-size integers

Tim Roughgarden

Proof Idea

Fix a comparison-based sorting method and an array length n.

$\Rightarrow$ consider input arrays containing $\{1,2,3,\ldots,n\}$ in some order. $\Rightarrow$ $n!$ such inputs

Suppose algorithm always makes $\leq k$ comparisons to correctly sort these $n!$ inputs.

$\Rightarrow$ across all $n!$ possible inputs, algorithm exhibits $\leq 2^k$ distinct executions

i.e., resolution of the comparisons

Proof Idea (con'd)

By the Pigeonhole Principle: if $2^k < n!$, execute identically on two distinct inputs $\Rightarrow$ must get one of them incorrect.

So: since method is correct, $2^k \geq n!$

$$\geq \left(\frac{n}{2}\right)^{\frac{n}{2}}$$

$$\Rightarrow k \geq \frac{n}{2}\log_2\frac{n}{2} = \Omega(n \log n)$$

QED!

Tim Roughgarden