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 : Merge Sort, Quick Sort, Heap Sort

Non Examples : Bucket Sort, Counting Sort, Radix Sort

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,…,n} in some order.

$\Rightarrow$ n! such inputs

Suppose algorithm always makes <= k comparisons to correctly sort these n! inputs.

=> Across all n! possible inputs, algorithm exhibits <= 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 => must get one of them incorrect.

So : Since method is correct,

$$
\begin{aligned}
2^k &\geq n! \\
&\geq \left(\frac{n}{2}\right)^{\frac{n}{2}}
\end{aligned}
$$

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