CS 401 Computer Algorithms I, UIC, Fall 2012
A Lower Bound on Sorting

Given a list A of n items to sort, a *comparison-based* sorting algorithm is one which accesses the input by performing queries of the form: *Is element* $A[i] \leq A[j]$ *for* $i \neq j$?

Claim 1. *Any (deterministic) comparison-based sorting algorithm requires at least* $\Omega(n \log n)$ *comparisons.*

Proof. To begin, note that sorting the input list A is equivalent to determining the current ordering on the n elements of A. In other words, we must identify which of the $n!$ possible orderings of n elements A is in. Observe now that each time we make a comparison query, e.g. "is $A[i] \leq A[j]$?", we eliminate half of the possible orderings of A. This is because in half of the remaining orderings $A[i]$ appears before $A[j]$, and in the other half of the remaining orderings $A[j]$ appears after $A[i]$. Hence, we conclude that in order to identify the ordering of A, we require $\Omega(\log(n!))$ comparisons.

To now simplify this latter expression, note first that $n! = n(n-1)(n-2)\cdots 2 \cdot 1 \leq n^n$. Conversely,

$$n! = n(n-1)(n-2)\cdots 2 \cdot 1 \geq n(n-1)(n-2)\cdots(n-\frac{n}{2}) \geq \left(\frac{n}{2}\right)^{\frac{n}{2}}.$$

Taking the logarithm of both these upper and lower bounds, we find $\log(n!) \in \Theta(n \log n)$. (Alternatively, one could also use Stirling's approximation to simplify $\Omega(\log(n!))$.) $\square$