34.4 Summary

Math Problem Out of Nowhere 1. We showed that $N! \in \Omega(\frac{N}{2}^{\frac{N}{2}})$.

Math Problem Out of Nowhere 2. We showed that $\log(N!) \in \Omega(N \log N)$, and that $N \log N \in \Omega(\log(N!))$. Therefore $\log(N!) \in \Theta(N \log N)$.

Seeking a Sorting Lower Bound. We've found a number of sorts that complete execution in $\Theta(N \log N)$ time. This raises the obvious question: Could we do better? Let TUCS (which stands for "The Ultimate Comparison Sort") be the best possible algorithm that compares items and puts them in order. We know that TUCS's worst case runtime is $O(N \log N)$ because we already know several algorithm whose worst case runtime is $\Theta(N \log N)$, and TUCS's worst case runtime is $\Omega(N)$ because we have to at least look at every item. Without further discussion, this analysis so far suggest that might be able to do better than $\Theta(N \log N)$ worst case time.

Establishing a Sorting Lower Bound. As a fanciful exercise, we played a game called puppy-cat-dog, in which we have to identify which of three boxes contains a puppy, cat, or dog. Since there are 3! = 6 permutations, we need at least $ceil(\lg(6)) = 3$ questions to resolve the answer. In other words, if playing a game of 20 questions with 6 possible answers, we have to ask at least 3 questions to be sure we have the right answer. Since sorting through comparisons is one way to solve puppy-cat-dog, then any lower bound on the number of comparisons for puppy-cat-dog also applies to sorting. Given N items, there are $N!$ permutations, meaning we need $lg(N!)$ questions to win the game of puppy-cat-dog, and by extension, we need at least $lg(N!)$ to sort N items with yes/no questions. Since $log(N!) = \Theta(N \log N)$, we can say that the hypothetical best sorting algorithm that uses yes/no questions requires $\Omega(N \log N)$ yes/no questions. Thus, there is no comparison based algorithm that has a worst case that is a better order of growth than $\Theta(N \log N)$ compares.

34.5 Exercises