Problem statement Questions about asymptotic growth "near" ∞ occur naturally when computer scientists analyze algorithms. One seemingly simple problem is sorting. How many comparisons are required to sort a list of n numbers? *Sorting and Searching*, volume 3 of *The Art of Computer Programming*, by D. Knuth, gives the following average running times for several sorting algorithms as a function of n:

Name	**Running time**
Comparison	$4n^2 + 10n$
Merge exchange	$3.7n(\ln n)^2$
Heapsort	$23.08n \ln n + 0.2n$

Which sorting method would you rather use if, in your application, $10 \leq n \leq 20$ (e.g., sorting a bridge hand)? Which would you rather use if $100 \leq n \leq 150$ (e.g., sorting grades in a lecture course)? Which would you rather use if $n \approx 10^6$ (e.g., sorting license plate numbers in New Jersey)? What happens to these functions as $n \to \infty$?