Big-Omega

- provides a way of saying a function grows at a rate 'greater than or equal to' another
- Let $f(n)$ and $g(n)$ be functions mapping integers to real #'s.

  We say $f(n)$ is $\Omega(g(n))$, if $g(n)$ is $O(f(n))$; that is, there is a constant $c > 0$ and int. constant $n_0 > 1$ such that:

$$f(n) \geq c\,g(n), \text{ for } n \geq n_0$$

Example:

$3n\log n - 2n$ is $\Omega(n\log n)$

- $3n\log n - 2n = n\log n + 2n(\log n - 1) \geq n\log n$ for $n \geq 2$; take $c=1$ and $n_0 = 2$

Big-Theta

- allows us to say 2 functions grow at the same rate
- We say $f(n)$ is $\Theta(g(n))$ if $f(n)$ is $O(g(n))$ and $f(n)$ is $\Omega(g(n))$.

  That is, there are real constants $c' > 0$, $c'' > 0$ and $n_0 > 1$ such that

$$c'g(n) \leq f(n) \leq c''g(n), \text{ for } n \geq n_0$$

Example:

$3n\log n + 4n + 5\log n$ is $\Theta(n\log n)$

- $3n\log n \leq 3n\log n + 4n + 5\log n \leq (3+4+5)\,n\log n$ for $n \geq 2$

Prefix Averages

- given a sequence S consisting of n numbers, we want to compute a sequence A such that $A[j]$ is the average of elements $S[0], \dots, S[j]$, for $j = 0, \dots, n-1$:

$$A[j] = \frac{\sum_{i=0}^{j} S[i]}{j+1}$$

- analyze three different implementations that solve this problem w/ vastly different running times