Design and Analysis
of Algorithms I

Asymptotic Analysis

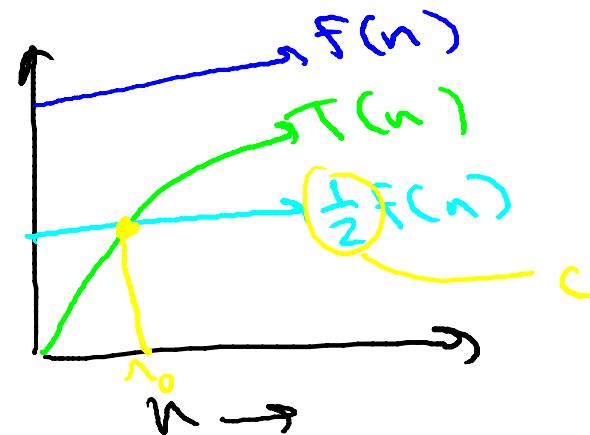## Big-Oh: Relatives (Omega & Theta)

Omega Notation

Definition : $T(n) = \Omega(f(n))$
If and only if there exist
constants c, n_0 such that

$$T(n) \geq c \cdot f(n) \quad \forall n \geq n_0 .$$

Picture

$$T(n) = \Omega(f(n))$$

Tim Roughgarden

Theta Notation

<u>Definition</u> : $T(n) = \theta(f(n))$ if and only if
$T(n) = O(f(n))$ and $T(n) = \Omega(f(n))$

<u>Equivalent</u> : there exist constants c_1, c_2, n_0 such that

$$c_1 f(n) \leq T(n) \leq c_2 f(n)$$

$$\forall n \geq n_0$$

Let $T(n) = \dfrac{1}{2}n^2 + 3n$. Which of the following statements are

true ? (Check all that apply.)

☐ $T(n) = O(n)$.

☐ $T(n) = \Omega(n)$. $\left[n_0 = 1, c = 4\right]$

☐ $T(n) = \Theta(n^2)$. $\left[n_0 = 1, c_1 = 1/2, c_2 = 4\right]$

☐ $T(n) = O(n^3)$. $\left[n_0 = 1, c = 1/2\right]$

Little-Oh Notation

Definition : $T(n) = o(f(n))$ if and only if for all constants c>0, there exists a constant n_0 such that

$$T(n) \leq c \cdot f(n) \quad \forall n \geq n_0$$

Exercise : $\forall k \geq 1, n^{k-1} = o(n^k)$

Where Does Notation Come From?

"On the basis of the issues discussed here, I propose that members of SIGACT, and editors of compter science and mathematics journals, adopt the O, Ω, and Θ notations as defined above, unless a better alternative can be found reasonably soon".

-D. E. Knuth, "Big Omicron and Big Omega and Big Theta", SIGACT News, 1976. Reprinted in "Selected Papers on Analysis of Algorithms."

Tim Roughgarden