



Design and Analysis  
of Algorithms I

# Asymptotic Analysis

---

## Big-Oh: Definition

# Big-Oh: English Definition

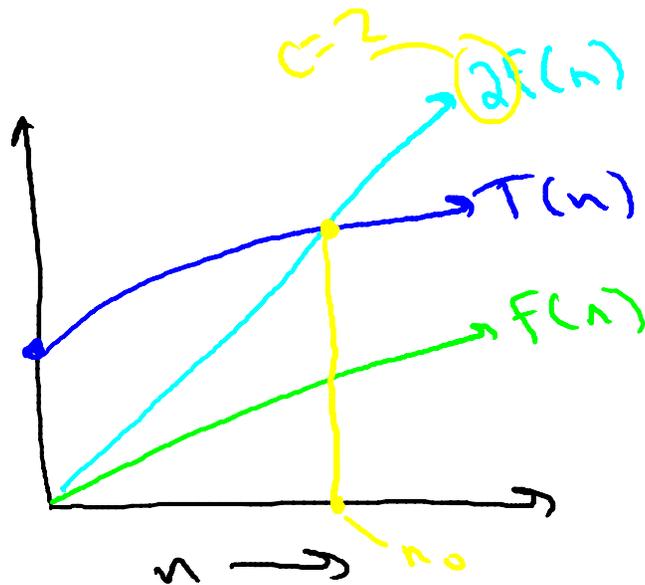
Let  $T(n)$  = function on  $n = 1, 2, 3, \dots$

[usually, the worst-case running time of an algorithm]

Q: when is  $T(n) = O(f(n))$ ?

A: if eventually (for all sufficiently large  $n$ ),  $T(n)$  is bounded above by a constant multiple of  $f(n)$ .

# Big-Oh: Formal Definition



Picture:  $T(n) = O(f(n))$

Formal Definition:  $T(n) = O(f(n))$

if and only if there exist constants  $c, n_0 > 0$  such that

$$\underline{T(n) \leq c \cdot f(n)}$$

for all  $n \geq n_0$ .

Warning:  $c, n_0$  cannot  
depend on  $n$ .