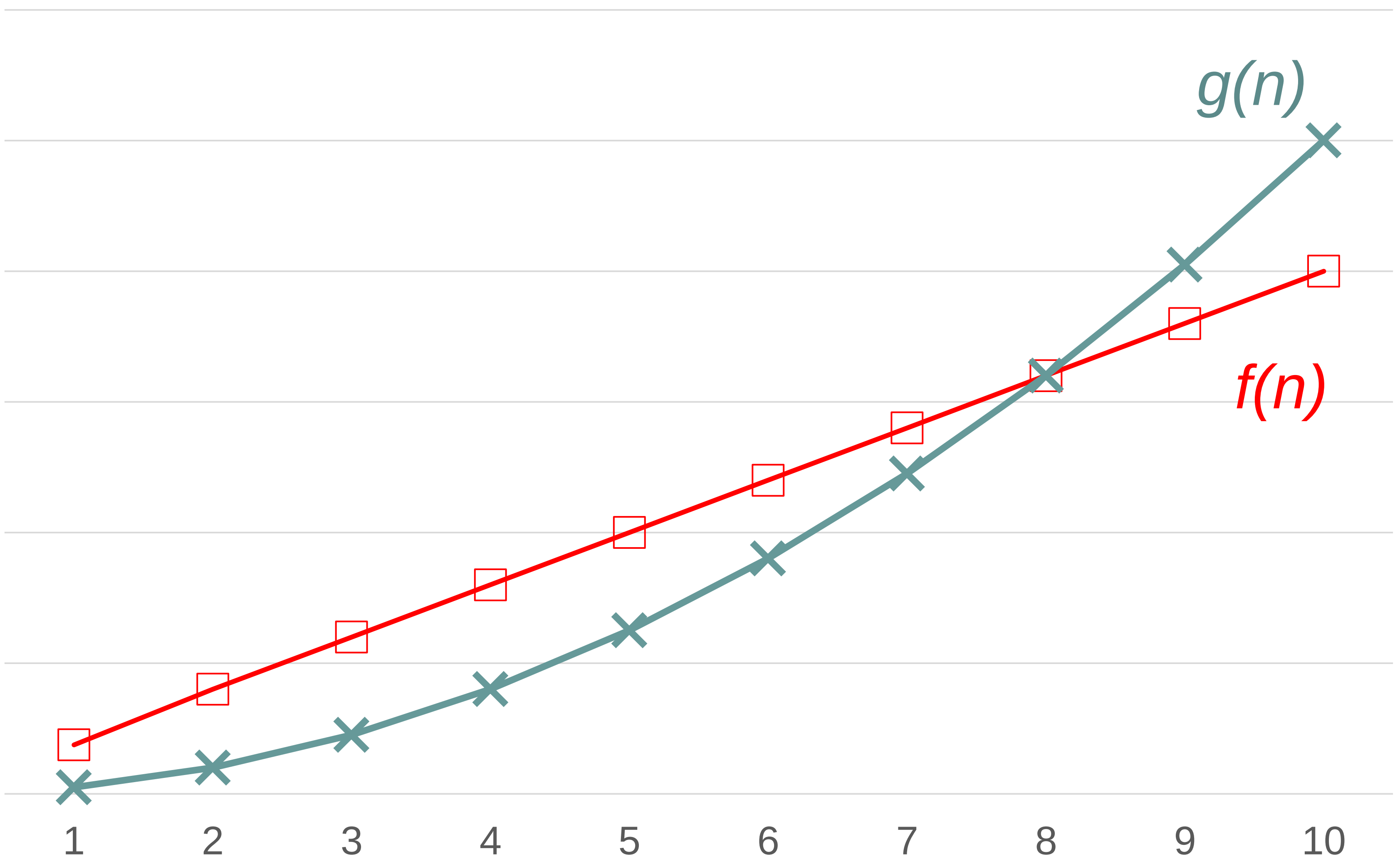
Growth of Functions

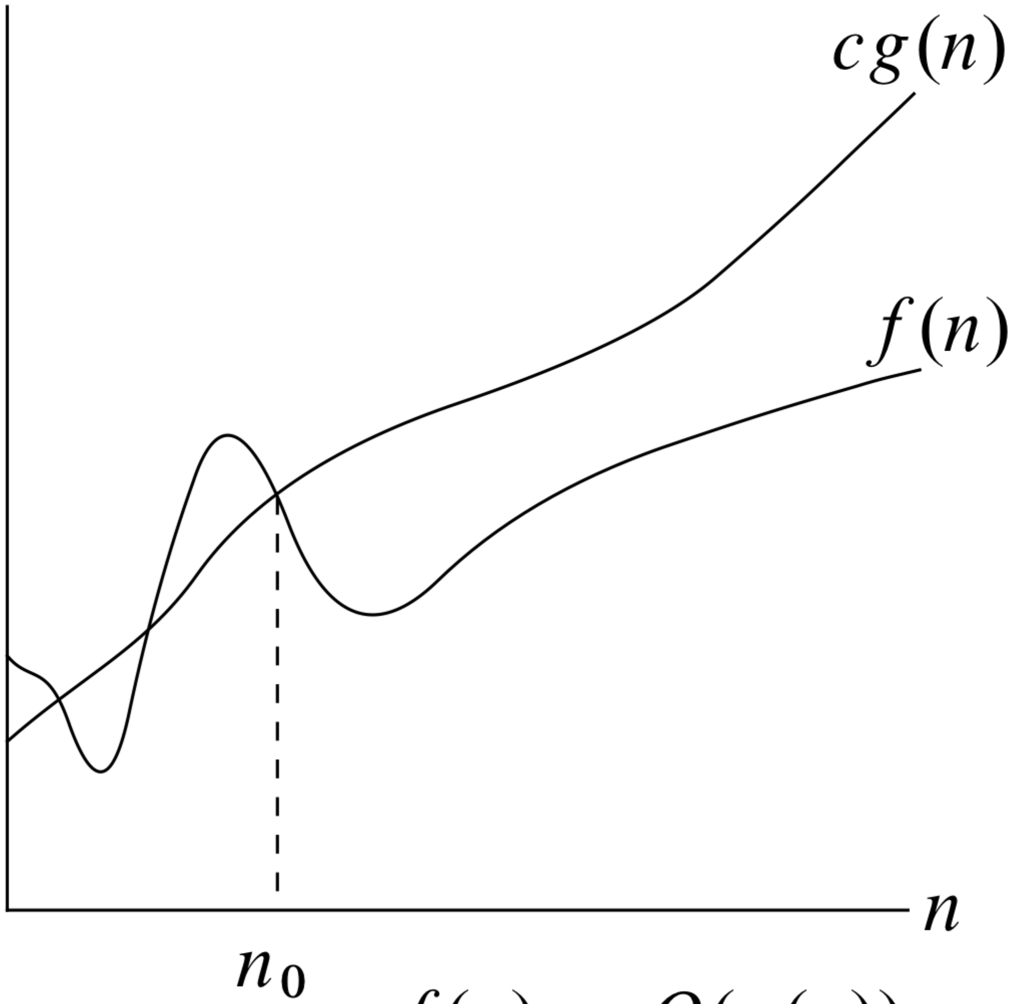
Learning Objectives

- Understand the meaning of growth of functions.
- Measure the growth of the running time of an algorithm.
- Use the Big-Oh notation to compare the growth of two functions.

Growth of Functions



O-notation

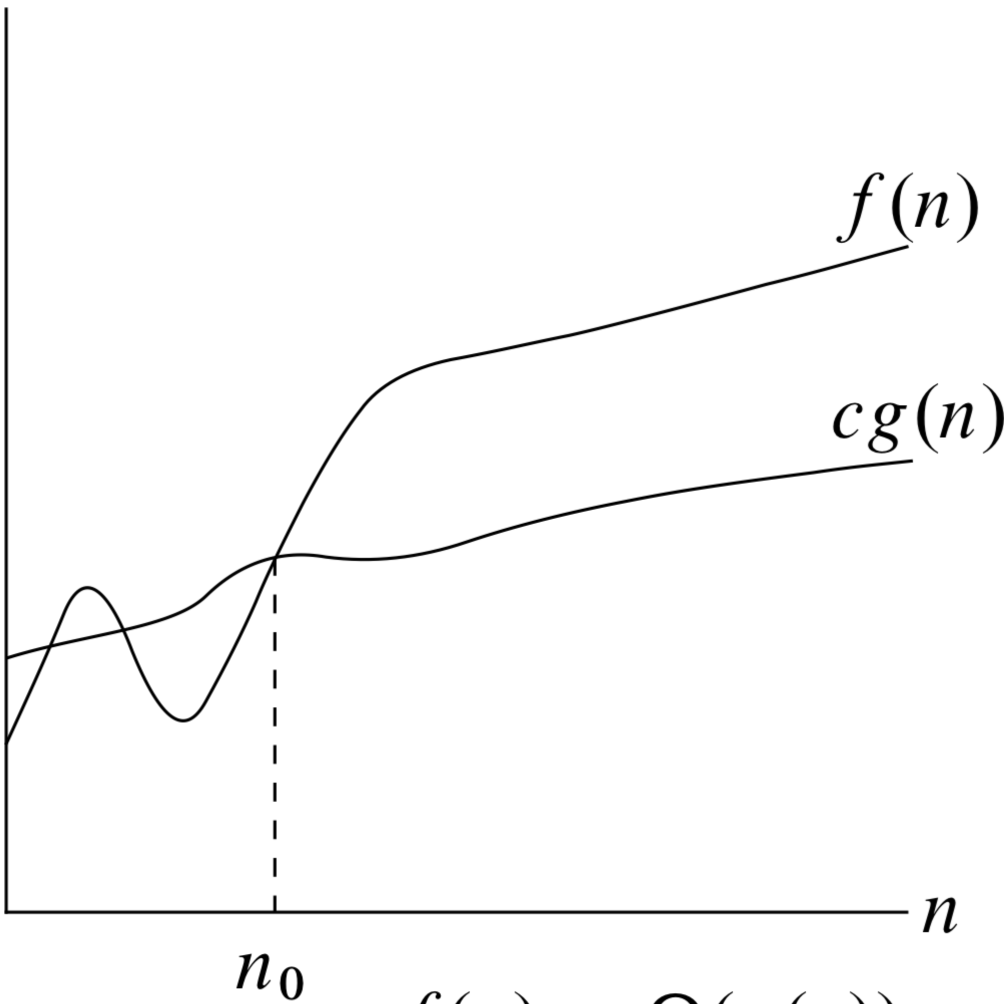


$$f(n) = O(g(n))$$

$$\exists c > 0, n_0 > 0$$
$$0 \le f(n) \le cg(n)$$
$$n \ge n_0$$

g(n) is an asymptotic upper-bound for f(n)

Ω-notation

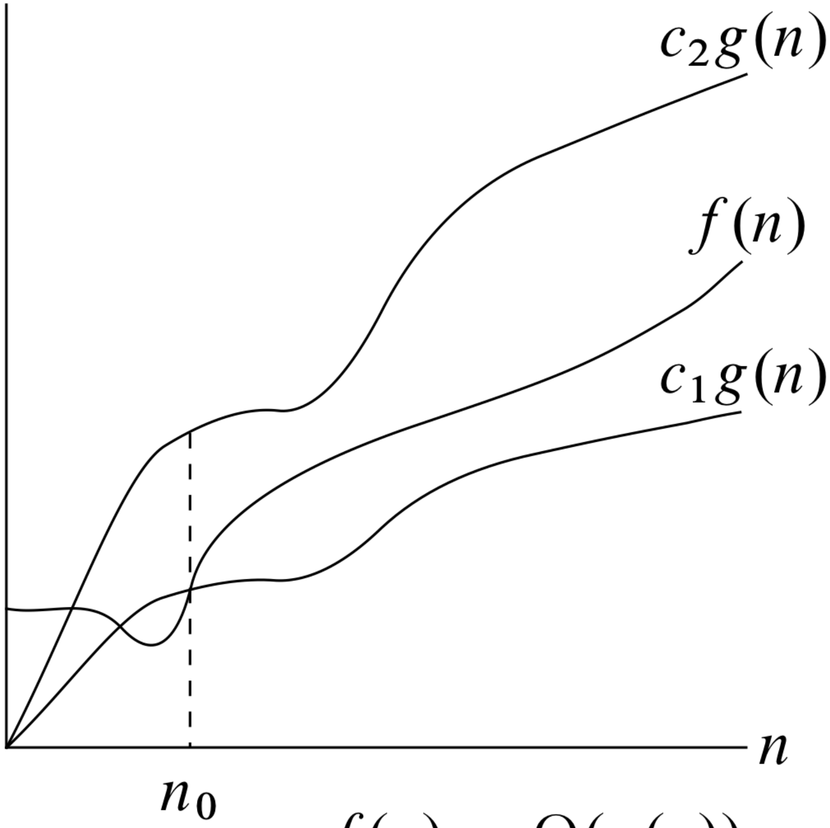


$$f(n) = \Omega(g(n))$$

$$\exists c > 0, n_0 > 0$$
$$0 \le cg(n) \le f(n)$$
$$n \ge n_0$$

g(n) is an asymptotic lower-bound for f(n)

Θ-notation

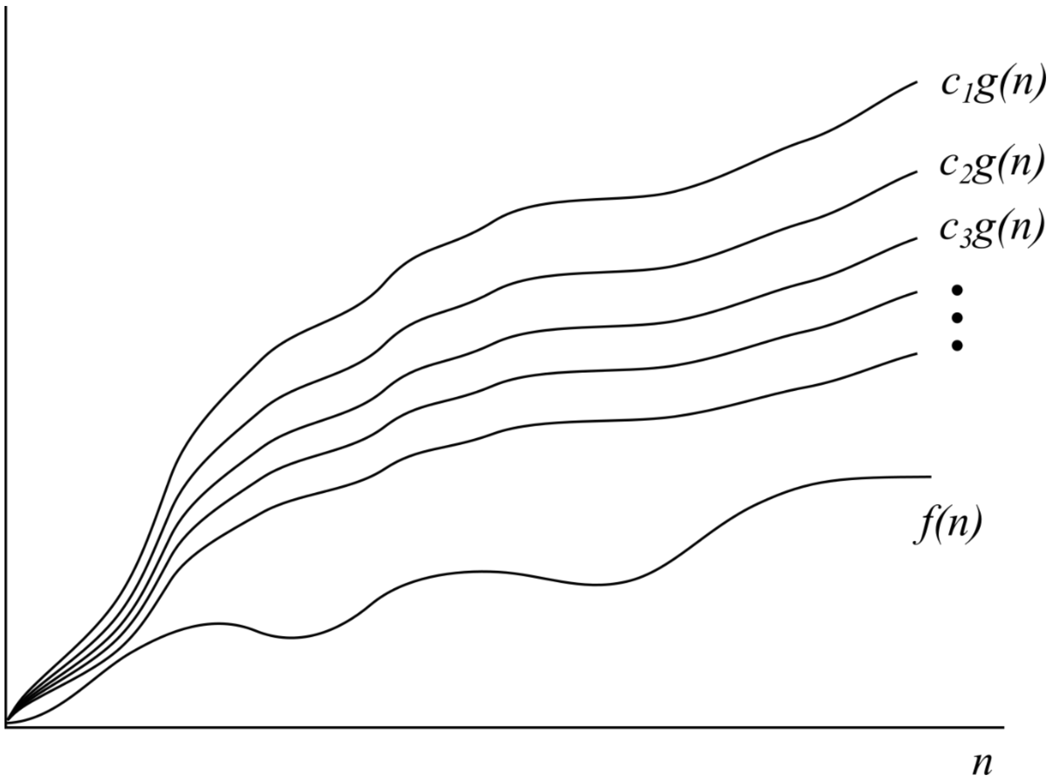


$f(n) = \Theta(g(n))$

$$\exists c_1, c_2 > 0, n_0 > 0$$
$$0 \le c_1 g(n) \le f(n) \le c_2 g(n)$$
$$n \ge n_0$$

g(n) is an asymptotic tight-bound for f(n)

o-notation

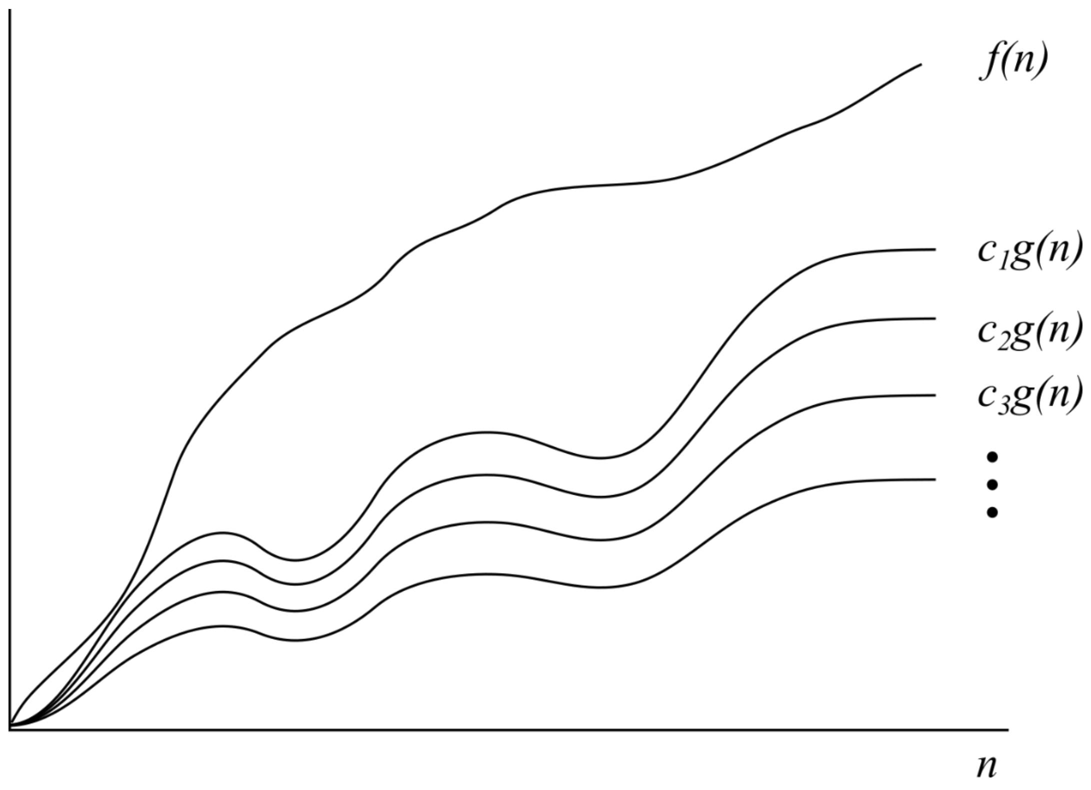


$$f(n) = o(g(n))$$

$$\forall c > 0$$
$$\exists n_0 > 0$$
$$0 \leq f(n) \leq cg(n)$$
$$n \geq n_0$$

g(n) is a non-tight asymptotic upper-bound for f(n)

ω-notation



$$f(n) = \omega(g(n))$$

$$\forall c > 0$$
$$\exists n_0 > 0$$
$$0 \leq cg(n) \leq f(n)$$
$$n \geq n_0$$

g(n) is a non-tight asymptotic lower-bound for f(n)

Analogy to real numbers

Functions	Real numbers
$\boldsymbol{f(n) = O\big(g(n)\big)}$	$a \leq b$
$\boldsymbol{f(n) = \Omega\big(g(n)\big)}$	$a \geq b$
$\boldsymbol{f(n) = \Theta\big(g(n)\big)}$	$a = b$
$\boldsymbol{f(n) = \mathrm{o}\big(g(n)\big)}$	$a < b$
$\boldsymbol{f(n) = \omega\big(g(n)\big)}$	$a > b$

Standard Classes of Functions

- Constant: $f(n) = \Theta(1)$
- Logarithmic: $f(n) = \Theta(\lg(n))$
- Sublinear: $f(n) = o(n)$
- Linear: $f(n) = \Theta(n)$
- Super-linear: $f(n) = \omega(n)$
- Quadratic: $f(n) = \Theta(n^2)$
- Polynomial: $f(n) = \Theta(n^k)$; k is a constant
- Exponential: $f(n) = \Theta(k^n)$; k is a constant

Insertion Sort (Revisit)

INSERTION-SORT(A, n)

for $j = 2$ **to** n
  $key = A[j]$
  // Insert $A[j]$ into the sorted sequence $A[1 \,..\, j-1]$.
  $i = j - 1$
  while $i > 0$ and $A[i] > key$
    $A[i+1] = A[i]$
    $i = i - 1$
  $A[i+1] = key$

j-times

n-times

$$\boxed{\Theta(n^2)}$$

Using L'Hopital's rule

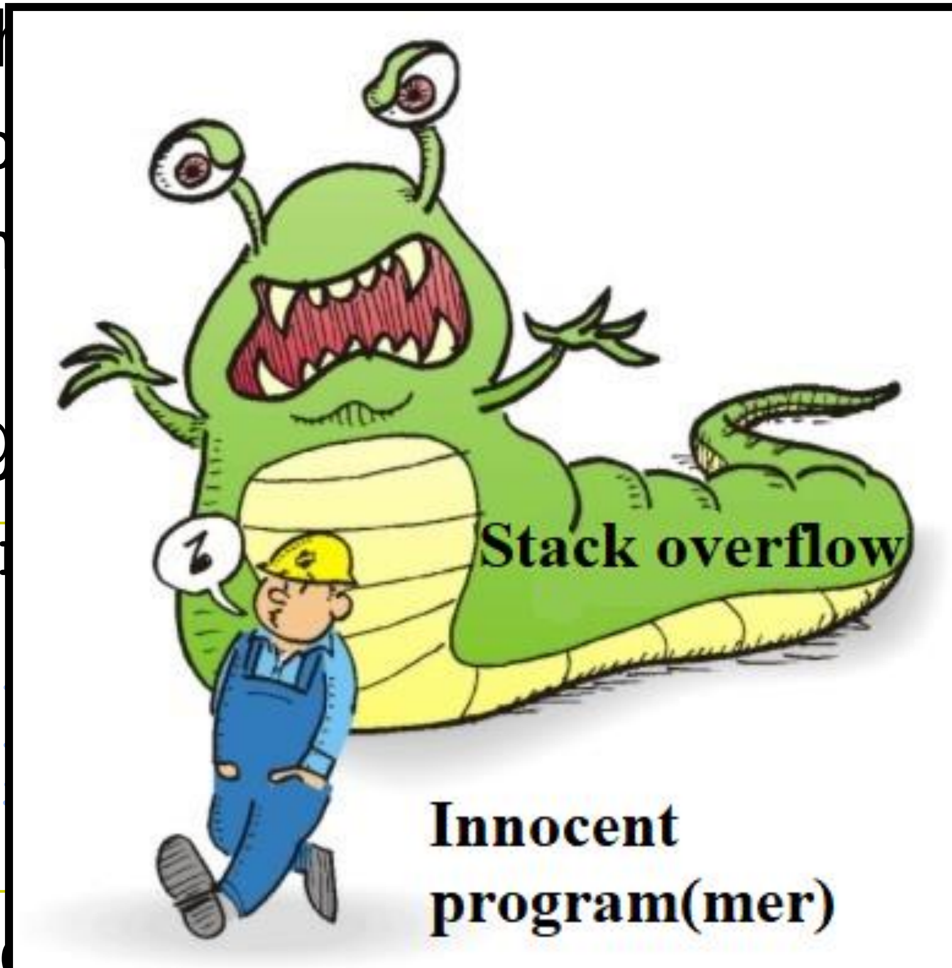
- Determine the relative growth rates by using L'Hopital's rule
  - compute $\lim_{n\to\infty} \frac{f(N)}{g(N)}$
  - if 0: f(N) = o(g(N))
  - if constant $\neq$ 0: f(N) = Θ(g(N))
  - if ∞: g(N) = o(f(N))
  - limit oscillates: no relation

Recursion

- In math [...] ed on itself
  - Facto
  - Fibon [...] $2), F(0) =$ $F(1)$
- In prog [...] self

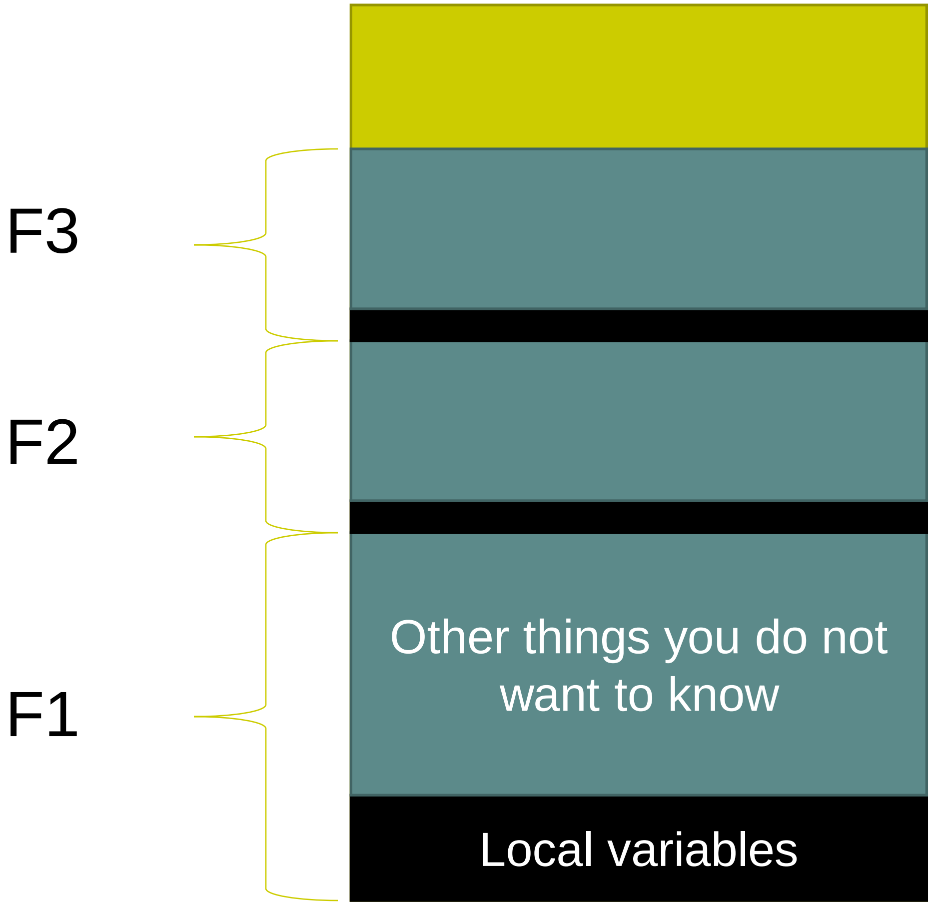
```
int fi
{


                                    ber-2);
}
```

- Questi [...] worst enemy?



Function calls

- main() {
    F1(…);
  }
- F1(…) {
    F2(…);
  }
- F2(…) {
    F3(…);
  }



Stack