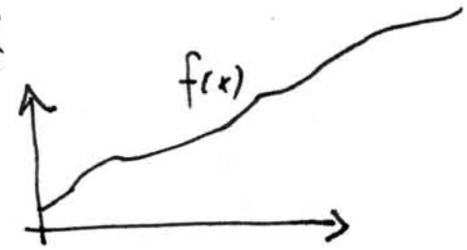
Asymptotic Notation for Functions



Functions: $f(n) \quad (\mathbb{N} \to \mathbb{R})$
$f(x) \quad (\mathbb{R} \to \mathbb{R})$

Goal: describe the asymptotic ($n \to \infty$) behavior (or growth) of $f(n)$

Intuition: $f(n) = O(g(n))$
"the growth of $f(n)$ is bounded by the growth of $g(n)$"

3 symbols:
O upper bound
Ω lower bound
Θ tight bound

Formal definitions

O = upper bound:

$$O(g(n)) = \{ f(n) \mid \text{there is } c > 0 \text{ and } n_0 \in \mathbb{N} \text{ such that } |f(n)| \leq c\,|g(n)| \text{ for } n \geq n_0 \}$$

Note: $O(g(n))$ is <u>a set</u> of functions

But: $f(n) \in O(g(n))$ is written as $f(n) = O(g(n))$
(abuse of notation)

Examples:
$n = O(n)$ since $n \leq 1 \cdot n$ for $n \geq 1$
$n = O(n^2)$ " $n \leq 1 \cdot n^2$ "
$n + \sqrt{n} = O(n)$ " $n + \sqrt{n} \leq 2n$ for $n \geq 1$

$\log_3 n = O(\log_2 n)$ $\quad \log_3 n \leq \log_2 n \quad n \geq 1$

$\log_2 n = O(\log_3 n)$ $\quad \log_2 n = \log_2 3 \cdot \log_3 n$

in general: $O(\log_a n) = O(\log_b n)$
so always just write $O(\log n)$

$\log n = O(n^\alpha)$ for $\alpha > 0$

since: $\lim_{x\to\infty} \frac{\log x}{x^\alpha} = \lim \frac{1/x}{x^{\alpha-1}} = 0$

Ω = lower bound:

$\Omega(g(n)) = \{f(n) \mid$ there is $c > 0$, $n_0 \in \mathbb{N}$ such that $|f(n)| \geq c|g(n)|$ for $n \geq n_0\}$

Examples: $n^2 = \Omega(n)$ (check!)

$n - 1000 = \Omega(n)$

Θ = tight bound:

$\Theta(g(n)) = \{f(n) \mid$ there is $c_1, c_2 > 0$, $n_0 \in \mathbb{N}$ such that $c_1|g(n)| \leq |f(n)| \leq c_2|g(n)|$, $n \geq n_0\}$

$= O(g(n)) \wedge \Omega(g(n))$

Example: $n^2 + n + 1 = \Theta(n^2)$

since $1 \cdot n^2 \leq n^2 + n + 1 \leq 3n^2$ for $n \geq 1$

Properties:

- $f(n) = \Theta(g(n)) \Rightarrow g(n) = \Theta(f(n))$
- $f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n)) \Rightarrow f(n) = \Theta(h(n))$
  (same for O)
- Careful:
  $f(n) = n$, $g(n) = n^{1+\sin n}$
  $f(n) \neq O(g(n))$, $g(n) \neq O(f(n))$

Other uses

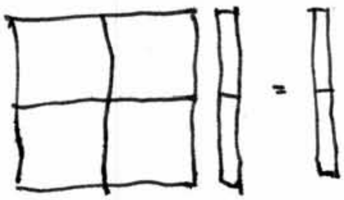
$n^2 + O(n) = O(n^2)$

means: for all $f(n) = O(n)$, $n^2 + f(n) = O(n^2)$

$\sum_{i=1}^{n} O(i) = O(n^2)$

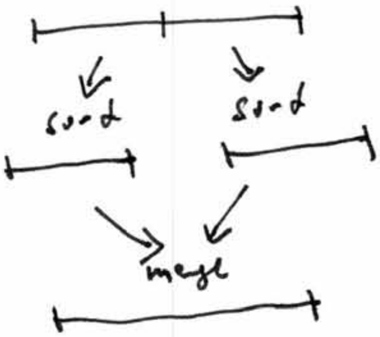
Divide-and-Conquer algorithms

1.) Matrix-vector multiplication (MVM)



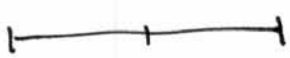
$\Rightarrow \Theta(n^2)$ (no gain)

2.) Merge sort



$\Rightarrow \Theta(n \log n)$

3.) Find element in sorted list



- cut in middle and compare
- proceed with right or left half

$\Rightarrow \Theta(\log n)$

4.) Karatsuba algorithm

Multiply polynomials with trick:

usual: $(a+bx)(c+dx) = ac + (ad+bc)x + bdx^2$

4 mults, 1 add

$= ac + ((a+b)(c+d) - ac - bd)x + bdx^2$

3 mults, 4 adds

$\Rightarrow \Theta(n^{\log_2 3})$