

# Lecture Feb 25

Monday, February 25, 2019 11:17 AM

## Asymptotic Notation

(first part ~ 4 min missed)

### Properties

1. Constants don't matter . For  $d > 0$ ,  $bf(n) \in O(f(n))$  and  $f(n) \in O(d f(n))$

2. Low order terms don't matter If  $\lim_{n \rightarrow \infty} \frac{h(n)}{g(n)} = 0$ , then  
 $g(n) + h(n) \in O(g(n))$  or  
 $g+h \in O(g)$

3 Transitivity If  $f(n) \in O(g(n))$  and  $g(n) \in O(h(n))$   
Then  $f(n) \in O(h(n))$

4. Summation rules If  $f_1(n) \in O(g_1(n))$  and  $f_2(n) \in O(g_2(n))$   
Then  $f_1(n) + f_2(n) \in O(g_1(n) + g_2(n))$

$$\max \{f(n), g(n)\} \in O(f(n) + g(n))$$
$$f(n) + g(n) \in O(\max \{f(n), g(n)\})$$

If  $f(n) \in O(g(n))$ , then  $\max \{f(n), g(n)\} \in O(g(n))$

5. Product Rule  $f_1(n) \times f_2(n) \in O(g_1(n) \times g_2(n))$

$\Omega$  - Notation :  $g \in \Omega(f)$  if and only if  $f \in \Omega(g)$

$\Theta$  - Notation :  $\Theta(f) = O(f) \cap \Omega(f)$  or

$$\{g \in \mathbb{F} \mid \exists c_1 \in \mathbb{R}^+, \exists c_2 \in \mathbb{R}^+, \exists b \in \mathbb{N}$$
$$n \geq b \text{ IMPLIES } c_1 f(n) \leq g(n) \leq c_2 f(n)\}$$

6 Logarithms and Exponents

Let  $a, b$  be positive constants

- If  $a \leq b$ . then  $n^a \in O(n^b)$

## b Logarithms and Exponents

- Let  $a, b$  be positive constants
- If  $a \leq b$ , then  $n^a \in O(n^b)$   
 $a > b$                $n^a \notin O(n^b)$
- If  $1 < a \leq b$ , then  $a^n \in O(b^n)$   
 $1 < b < a$ , then  $a^n \notin O(b^n)$

For all  $a, b > 1$ .  $\log_a(n) \in O(\log_b(n))$

Convention: omit base (i.e.  $\log n$ )

7. Exponential Function grow  
faster than polynomials

For all  $a$  and  $b > 1$ :  
 $n^a \in O(b^n)$  and  $b^n \notin O(n^a)$

8 Polynomials grow faster than  
polylogarithmic functions

For all  $a, b > 0$ .  
 $(\log n)^a \in O(n^b)$  and  $n^b \notin O((\log n)^a)$

## Solving Recurrences

A recurrence is an inductively (recursively) defined function.

Solving an recurrence is finding a closed form for the function  
that is not recursive.

1. Guess and verify: generate a table of values for the function  
and look for pattern

2. Repeated Substitution  
and Verification

(Plug and chug)

$$\text{Example } \text{For } n \in \mathbb{Z}^+, M(n) = \begin{cases} c & \text{if } n=1 \\ M(\lfloor \frac{n}{2} \rfloor) + M(\lceil \frac{n}{2} \rceil) + dn & n>1 \end{cases}$$

If  $c=d=0$ , then  $M(n)=0$ .

Consider special case ·  $n$  is power of 2

$$\text{Then } M(n) = 2M\left(\frac{n}{2}\right) + dn$$

$$= 2 \left[ 2M\left(\frac{n}{2^2}\right) + \frac{dn}{2} \right] + dn$$

$$= 2^2 M\left(\frac{n}{2^2}\right) + 2dn$$

⋮

$$= 2^i M\left(\frac{n}{2^i}\right) + 2dn$$

⋮

$$= 2^i M\left(\frac{n}{2^i}\right) + i(dn)$$

When  $i = \log_2(n)$ . Then  $M(n) = n M(1) + (\log_2 n) dn$

$$= cn + dn \log_2 n$$

When trying to prove by induction, Let:

$$Q(k) = "M(2^k) = c2^k + dk2^k"$$

$\forall k \in \mathbb{N} \quad Q(k).$