Yufei Cui

CSCA48 Tut 9 Notes (Provided by Dr. Marzieh & Cheng)

Complexity:

Q: Why do we need to analyze algo?
A: Find how efficient they are.
how much resources (i.e. cpu time) is required

Q: How do we analyze algo?
A: 1. Experimental : time consuming and does not provide proper comparison :(
2. Theoretical: Count # of operation required (more rigorous) :)

Q: why do we need big oh?
A: Simple answer: 1) allow us to ignore constant fact.
2) " " ignore lower order terms
3) only focus on dominating term (which is what matters asymptotically)

Defn:
$f \in O(g)$ ~ set iff

$\exists\, c > 0,\ \exists\, n_0 > 0$ s.t. if $n \geq n_0$ then $f(n) \leq c \cdot g(n)$

where $f, g : \mathbb{R}^+ \to \mathbb{R}^+$ (usually, we only consider $\mathbb{N} \to \mathbb{R}^+$)

Note: $O(g)$ is a set of functions

$O(g) : \{ f \mid \exists\, c>0, \exists\, n_0 > 0$ s.t. if $n \geq n_0$ then $f(n) \leq c \cdot g(n) \}$

$\Rightarrow$ writing $f = O(g)$ is a sloppy notation (func $\neq$ set)
use $f \in O(g)$ [f is a member of $O(g)$]

Exercises:

Prove: $5n^4 + 3n^2 + 2n^2 + 4n + 1 \in O(n^4)$

pf: Choose $C =$ 15 (or larger)
Choose $n_0 =$ 1 (or larger)

Consider LHS $= 5n^4 + 3n^2 + 2n^2 + 4n + 1$
$\leq 5n^5 + 3n^5 + 2n^5 + 4n^5 + n^5$ (True $\forall n \geq 1$)
$= 15n^5$
$= Cn^5$

$\boxtimes$

Note: if $n \geq 1$ then $1 \leq n \leq n^2 \leq n^3 \leq \ldots \leq n^d$ [*]

prove: if $f(n) = a_0 + a_1 n + a_2 n^2 + \ldots + a_d n^d$
$g(n) = n^d$
then $f \in O(g)$.

pf: Choose $C = \sum_{i=0}^{d} |a_i|$
Choose $n_0 =$ 1 (or larger)

Consider $a_0 + a_1 n + \ldots + a_d n^d$
$\leq a_0 n^d + a_n n^d + \ldots + a_d n^d$ [by *]
$= n^d (a_0 + a_1 + \ldots + a_d)$
$\leq n^d (|a_0| + |a_1| + \ldots + |a_d|)$
$\leq C \cdot n^d$

$\boxtimes$

Prove: $10n^3 + 5n^2 + 3n\log n + 1 \in O(n^3)$ [log base 10]

Choose $c = \underline{\ 19\ }$

$n_0 = \underline{\ 1\ }$

Consider $10n^3 + 5n^2 + 3n\log n + 1$

$\leq 10n^3 + 5n^3 + 3n^3 + n^3$ [by * and $\log n \leq n^2$ $\forall n \geq 1$]

$= 19n^3$

$= c \cdot n^3$

$\boxtimes$

Prove: $3\log n + 2 \in O(\log n)$

Choose $c = \underline{\ 5\ }$

$n_0 = \underline{\ 10\ }$

Consider $3\log n + 2$

$\leq 3\log n + 2\log n$ [$\forall n \geq 10$, $\log n \geq 1$]

$= 5\log n$

$= c \log n$

$\boxtimes$

Prove $2^{n+2} \in O(2^n)$

Choose $c = \underline{\ 4\ }$

$n_0 = \underline{\ 1\ }$

Consider 2^{n+2}

$= 2^n \cdot 2^2$ [exp rule]

$= 4 \cdot 2^n$

$= c \cdot 2^n$

$\boxtimes$

We care mostly about the following order of fncs –

$f_1 = 1$, $f_2 = \log n$, $f_3 = n$, $f_4 = n \log n$, $f_5 = n^2$, $f_6 = 2^n$

and
$f_1 \in O(f_2)$
$f_2 \in O(f_3)$
$f_3 \in O(f_4)$
$f_4 \in O(f_5)$
$f_5 \in O(f_6)$

where does $f = \log(\log n)$ fit in here?

Exercise: Prove the following

Sum property: if $f_1 \in O(g_1)$
$f_2 \in O(g_2)$
then $f_1 + f_2 \in O(\max(g_1, g_2))$

product property: if $f_1 \in O(g_1)$
$f_2 \in O(g_2)$
then $f_1 \cdot f_2 \in O(g_1 \cdot g_2)$

transitivity property: if $f \in O(g)$
$g \in O(h)$
then $f \in O(h)$