37181 DISCRETE MATHEMATICS

Lecture 11: Big O, complexity of algorithms

PLAN

- Big O
- comparing speed of algorithms

COMPARING SPEEDS OF ALGORITHMS

for i in [1, n]

```
power_slow(a real; n positive integer)
x = 1.0
for i = 1 to n do
    x = x*a
return x
```

i	x
	1.0
1	a
2	$a * a = a^2$
3	$a^2 * a = a^3$
4	$a^3 * a = a^4$
$\vdots$	
n	a^n

~~$L(i, x)$:~~

~~$x = a^i$~~

COMPARING SPEEDS OF ALGORITHMS

```
power_slow(a real; n positive integer)
x = 1.0
for i = 1 to n do
    x = x*a
return x
```

one iteration

power_slow(3, 6):

i	x
1	3
2	3^2
3	
4	
5	
6	3^6

Question

how many steps?

iterations of for loop $= 6$.

```
power_slow(a real; n positive integer)
x = 1.0
for i = 1 to n do
    x = x*a
return x
```

m

power_slow(3, 100):

i	x
	1.0
1	
2	
100	

100 steps.

COMPARING SPEEDS OF ALGORITHMS

```
power_fast(a real; n positive integer)
x = 1.0
i = n
while i > 0 do
    if i is odd then
        x = x*a
    i = floor(i/2)
    if i > 0 then
        a = a*a
return x
```

i	x	a
n	1.0	a
$\lfloor \frac{n}{2} \rfloor$		

A "step" of this algorithm is one iteration of the while loop.

COMPARING SPEEDS OF ALGORITHMS

```
power_fast(a real; n positive integer)
x = 1.0
i = n
while i > 0 do
    if i is odd then
        x = x*a
    i = floor(i/2)
    if i > 0 then
        a = a*a
return x
```

One step

power_fast(3, 6):

i	x	a
6	1.0	3
3	1.0	9
1	9	81
0	9·81 = 3^6	81

: 3^4

3^6

If input $n = 2^k$

```
power_fast(a real; n positive integer)
x = 1.0
i = n
while i > 0 do
    if i is odd then
        x = x*a
    i = floor(i/2)
    if i > 0 then
        a = a*a
return x
```

i	x	a
100	1.0	3
50		
25		
12		
6		
3		
1		
0		

power_fast(3, 100):

$100 = 2^6 + 2^5 + 2^2$

64

36

If $n = 2^k$ $\quad k > 0$

$\frac{2^6}{2} = 2^5$

i	x	a
2^k		
2^{k-1}		
2^{k-2}		
$\vdots$		
2^1		
2^0		
0		

2^{k-k}

1

\# steps?

k

$\log_2 n = \log_2(2^k)$
$= k$

$k = \log_2 n$

ignoring size of a.

How can we *formally* capture the idea that the second algorithm is *faster*? Can we estimate that on input of size n, the algorithm will take roughly some function of n steps?

eg $\log_2 n$

Look back at the two algorithms and try to roughly guess a function for each one.

$\text{power\_fast}(a, 2^k) \to \log_2 k$

$\text{power\_fast}(a, 1{,}000{,}000)$

$2^{\square} + 2^{\square} + 2^{\square}$

$f_1(n) = n$
$f_2(n) = \log_2 n$

Luckily, someone already thought of how to capture the idea of algorithm speed/complexity, using this definition.

time

# of steps

stronger.

Definition

Let $f, g : \mathbb{N}_+ \to \mathbb{R}$. We say that g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that

$m = 1000$

$$|f(n)| \leqslant m|g(n)|$$

absolute value

for all $n \in \mathbb{N}, n \geqslant k$.

$(k=6)$

$n =$ 1 2 3 4 5 6 7

$f(n)$:

could be bigger

$|f(6)|$ $|f(7)|$

$\leq 1000|g(6)|$ $1000|g(7)|$

$g(n)$:

f is dominated by g.

if

$\exists\ m \in \mathbb{R}_+,\ k \in \mathbb{N}_+$

such that

$\forall\ n \geq k$

$$|f(n)| \leq m\,|g(n)|$$

COMPARING SPEEDS OF ALGORITHMS

Luckily, someone already thought of how to capture the idea of algorithm speed/complexity, using this definition.

Definition

Let $f, g : \mathbb{N}_+ \to \mathbb{R}$. We say that g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that

$$|f(n)| \leqslant m|g(n)|$$

for all $n \in \mathbb{N}, n \geqslant k$. "in Big O of g".

We use the notation $f \in O(g)$ and read this as "f is in Big O of g".

in

$O(g) = \{ f : \mathbb{N}_+ \to \mathbb{R} \mid \ldots$

Luckily, someone already thought of how to capture the idea of algorithm speed/complexity, using this definition.

Definition

Let $f, g : \mathbb{N}_+ \to \mathbb{R}$. We say that g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that

$$|f(n)| \leqslant m|g(n)|$$

for all $n \in \mathbb{N}, n \geqslant k$.

We use the notation $f \in O(g)$ and read this as "f is in Big O of g".

$O(g)$ is the set of all functions from $\mathbb{N}_+$ to $\mathbb{R}$ which are dominated by g.

whom

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}, n \geqslant k$.

Ex: who dominates who: $f(n) = \log_2 n + 5$ versus $g(n) = n$:

small ?

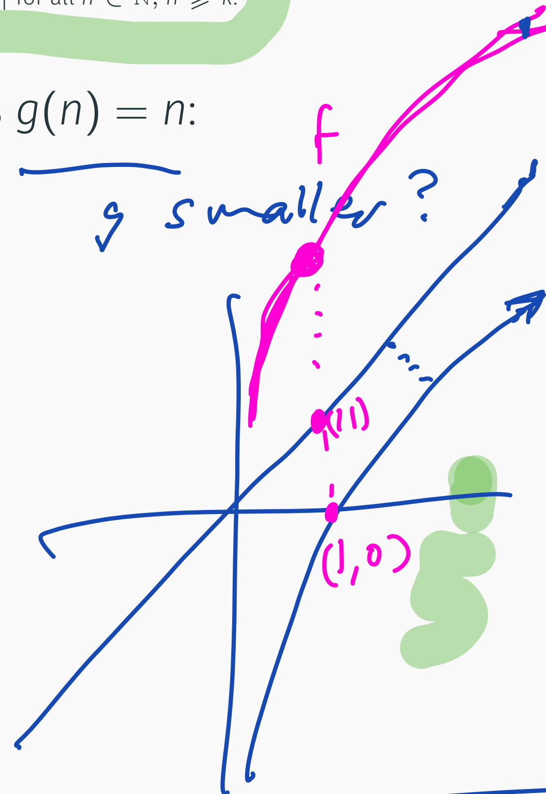
g smaller ?

Guess: $f \in O(g)$?

$\exists\, m = \boxed{2}$ $\quad \exists\, k = \boxed{5}$

$\forall\, n \geqslant \boxed{5}$

$\log_2 n + 5 \leq 2 \cdot n$



START HERE

if $n \geqslant 5$

$\log_2 n + 5 \le \log_2 n + n$

I need

$\log_2 n \le n$?

$e^{\log_2 n}$

iff (~~to~~ raise b.s power 2.)

$n = 2^{\log_2 n} \le 2^n$

START AGAIN

$p(n)$: $n \le 2^n$

Prove by induction

$n = 1$ $p(1)$: $1 \le 2^1 = 2$ ✓

Assume $p(k)$ true

$p(k+1)$: $k+1 \le k+k$

$= 2k$

$\le 2 \cdot 2^k$ by Ind. hyp.

$= 2^{k+1}$.

by PMI true $\forall n \geq 1$.

$n \le 2^n$ — assume $n \geq 1$

$\log_2 n \le n$

$\log_2 n + 5 \le n + n$ (assume $n \geq 5$)

$\le 2n$

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}, n \geqslant k$.

Ex: who dominates who: $f(n) = \log_2 n + 5$ versus $g(n) = n$:

Here is my prepared solution in case my "live" one is too messy:

Claim 1: $n \leqslant 2^n$. Let $P(n)$ be the statement $n \leqslant 2^n$. $P(1)$ true. Assume $P(k)$ then $n + 1 \leqslant n + n = 2n \leqslant 2.2^n = 2^{n+1}$ so $P(k+1)$ is implies, so by PMI true for all $n \geqslant 1$.

Take $\log_2$ both sides gives $\log_2 n \leqslant n$, so

$$\log_2 n + 5 \leqslant n + 5 \leqslant n + n$$

(if $n \geqslant 5$)

$$= 2n$$

so $m = 2, k = 5$ gives the result.

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}, n \geqslant k$.

Some careful analysis of the two algorithms above shows that, up to some constants, power_fast takes about $\log_2 n$ +5 steps (because each step divides i by 2, roughly) and power_slow takes ~~n~~ steps.

Because of Big O, if you count steps slightly differently, and/or count the initial and final lines of the code, the time complexity function changes a bit but is **the same** up to Big O.

Reason for m:

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}, n \geqslant k$.

Recall the formula

$$\log_b y = \frac{\log_a y}{\boxed{\log_a b}}.$$ [1]

Constant.

$$y = a^{\log_a y}$$

$$\log_b y = \log_b a^{(\log_a y)}$$

$$= \log_a y \, \log_b a$$

[1] Proof: $y = b^{\log_b y}$, so sub this into the LHS.

RECALL

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}$, $n \geqslant k$.

Recall the formula

$$\log_b y = \frac{\log_a y}{\log_a b}.$$ [1]

If $b = 2$ and your calculator only has a button for $\log_{10}$ or $\log_e = \ln$ then you can use this formula:

$$\log_2 n = \frac{\log_e n}{\log_e 2}.$$

$$\boxed{\frac{1}{\log_e 2}}$$

just a constant.

[1] ~~Proof~~ Hint: $y = a^{\log_a y}$, so sub this into the LHS.

$\log_7 x \in O(\log_{17} x)$
$\log_{17} x \in O(\log_7 x)$

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}$, $n \geqslant k$.

Recall the formula

$$\log_b y = \frac{\log_a y}{\log_a b}.^{1}$$

If $b = 2$ and your calculator only has a button for $\log_{10}$ or $\log_e = \ln$ then you can use this formula:

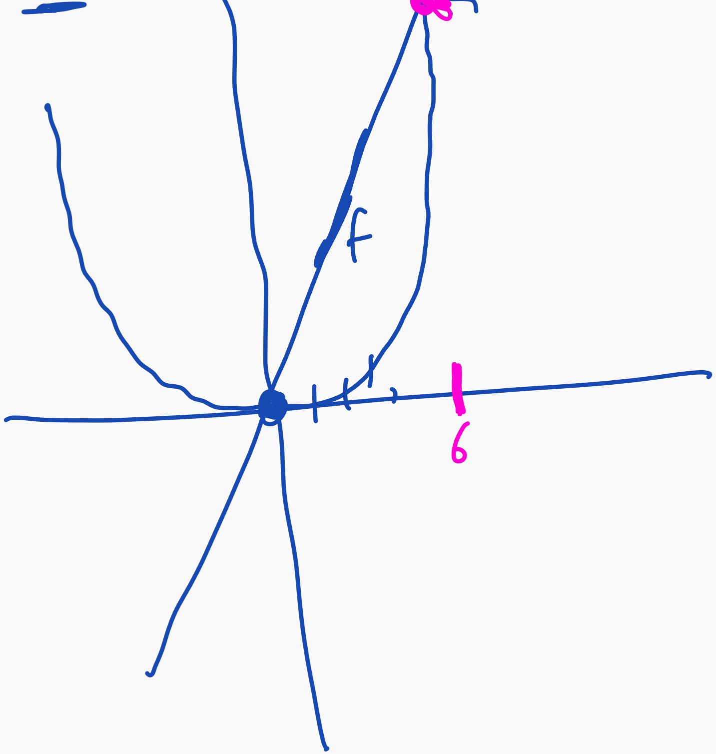
$$\log_2 n = \frac{\log_e n}{\boxed{\log_e 2}}.$$

Since $\log_e 2 \approx 4.6$ is a fixed number, it doesn't matter which log we use because of the m in the definition.

[1] Proof: $y = a^{\log_a y}$, so sub this into the LHS.

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}, n \geqslant k$.

Let $f(n) = 6n$ and $g(n) = n^2$. Show that g dominates f, that is, $6n \in O(n^2)$.



$\exists\ k = \boxed{6}$

$m = \boxed{1}$

$\forall\ n \geqslant 6$

$6 \cdot n \leq n \cdot n$ since $n \geqslant 6$

$= 1 \cdot n^2$

so $6n \in O(n^2)$ $\square$

Another proof

$\exists\ \underline{k = 1}\ ,\ \underline{m = 6}$

for all $n \geq \underbrace{1}_{k}$,

$6n = 6n \cdot 1 \leq 6n \cdot n$ since $n \geq 1$

$= \underbrace{6}_{m} n^2$

$\boxed{\text{So } 6n \in O(n^2).}$

EG

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}, n \geqslant k$.

Let $f(n) = 6n$ and $g(n) = n^2$. Show that g dominates f, that is, $6n \in O(n^2)$.

Here is my prepared solution. It might be different that the one I just did “live” – many choices for m, k can work.

There exist $m = 1, k = 6$ so that

$$6n \leqslant n.n = n^2$$

for all $n \geqslant 6$.

EG

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}$, $n \geqslant k$.

Show that $g(n) = n^2$ is not dominated by $f(n) = 6n$. That is, $n^2 \notin O(6n)$.

m

negation

EG

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}, n \geqslant k$.

Show that $g(n) = n^2$ is not dominated by $f(n) = 6n$. That is, $n^2 \notin O(6n)$.

We need the *negation* of the Big O definition.

$\neg (n \geq k) \vee (|f(n)| \leq m|g(n)|)$

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}, n \geqslant k$.

Show that $g(n) = n^2$ is not dominated by $f(n) = 6n$. That is, $n^2 \notin O(6n)$.

We need the *negation* of the Big O definition.

$$\forall m \in \mathbb{R}_+ \forall k \in \mathbb{N}_+ \exists n \in \mathbb{N}_+ [(n \geqslant k) \wedge (|f(n)| > m|g(n)|)].$$

$n^2 \notin O(6n)$

$\forall m \in \mathbb{R}_+ \quad \forall k \in \mathbb{N}_+$

$\exists n \qquad n \geq k, \quad \text{and} \quad n^2 > m \cdot 6n$

Choose ~~$n = \max\{k, m\}$.~~

Proof

$n^2 \notin O(6n)$.

since $\forall m \in \mathbb{R}_+$, $\forall k \in \mathbb{N}_+$

$\exists\ n = \boxed{\max\{k, 6m+1\}}$.

so that ① $n \geq k$

and ② $n^2 \geq (6m+1)n$ → since $n \geq 6m+1$

strictly $> 6m \cdot n$ $\qquad \max\{6m, k\}$

EG

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}$, $n \geqslant k$.

Show that $g(n) = n^2$ is not dominated by $f(n) = 6n$. That is, $n^2 \notin O(6n)$.

We need the *negation* of the Big O definition.

$$\forall m \in \mathbb{R}_+ \forall k \in \mathbb{N}_+ \exists n \in \mathbb{N}_+ [(n \geqslant k) \wedge (|f(n)| > m|g(n)|)].$$

Here is my prepared solution.

Given m, k fixed numbers (positive), there is a number $n = \max\{6m + 1, k\}$ so that $n > 6m$ so

$$n^2 > m6n.$$

$f \in O(g) \wedge g \in O(f)$

$\Rightarrow f \neq g.$

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}$, $n \geqslant k$.

Let $f(n) = 6n^2 + 5n + 2$ and $g(n) = n^2$. Show that $f \in O(g)$ and $g \in O(f)$. So they are (up to Big O equivalence) the "same".

refl: $f \in O(f)$

$|f(n)| \leq |f(n)|$

$\forall n \geqslant 1$

$m = 1$

$k = 1$

Define

$f \cancel{R} g$

if $f \in O(g)$.

Then $\cancel{R}$ is not antisymmetric.

$n^2 \le 6n^2$ $\quad 1 \le 6$

$\le 6n^2 + 5n + 2$ $\quad 5n \ge 5$

$5n+2 \ge 7$

≥ 0

$m = 1 \quad c^2 = 1$

$6n^2 + 5n + 2 \le 6n^2 + n^2 + 2$

since

$n \ge 5$

$\le 6n^2 + n^2 + n$ $\quad n \ge 2$

$\le 6n^2 + n^2 + n^2$ $\quad n \ge 1$

$= 8n^2$

$k = 5$

$m = 8$

EG

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}$, $n \geqslant k$.

Show that $f(n) = n^3$ is dominated e^n.

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}$, $n \geqslant k$.

Show that $f(n) = n^3$ is dominated e^n.

My prepared solution:

Let $P(n)$ be the statement $n^3 \leqslant e^n$. Need to find a value k so that $P(k)$ is true, then

assume for $s \geqslant k$ that $P(s)$ is true, then $P(s+1)$:

$$(n+1)^3 = n^3 + 3n^2 + 3n + 1 \leqslant n^3 + n^3$$

(by a Lemma I will prove below)

$$= 2n^3 \leqslant 2e^n < e.e^n = e^{n+1}$$

since $e = 2.7... > 2$.

Lemma: $3n^2 + 3n + 1 \leqslant n^3$ for n big enough.

CHALLENGE QUESTION

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}, n \geqslant k$.

Show that $f(n) = n^c$ is dominated e^n where c is any positive integer.

So polynomials are "slower" than exponentials.

CHALLENGE QUESTION

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}$, $n \geqslant k$.

Show that $f(n) = n^c$ is dominated e^n where c is any positive integer.

So polynomials are “slower” than exponentials.

Your proof might use the Binomial Theorem:

$$(x+y)^c = x^c + \binom{c}{1}x^{c-1}y + \binom{c}{2}x^{c-2}y^2 + \cdots + \binom{c}{c-1}x^1y^{c-1} + y^c$$

CHALLENGE QUESTION

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}$, $n \geqslant k$.

Show that $f(n) = n^c$ is dominated e^n where c is any positive integer.

So polynomials are "slower" than exponentials.

Your proof might use the Binomial Theorem:

$$(x+y)^c = x^c + \binom{c}{1}x^{c-1}y + \binom{c}{2}x^{c-2}y^2 + \cdots + \binom{c}{c-1}x^1y^{c-1} + y^c$$

$$(n+1)^c = n^c + \binom{c}{1}n^{c-1} + \binom{c}{2}n^{c-2} + \cdots + \binom{c}{c-1}n + 1$$

In your proof, c is a fixed number, so all the $\binom{c}{i}$ terms are bounded above by some fixed number.

Defn: g *dominates* f if there exist constants $m \in \mathbb{R}^+$ and $k \in \mathbb{N}_+$ such that $|f(n)| \leqslant m|g(n)|$ for all $n \in \mathbb{N}$, $n \geqslant k$.

Which is faster (who dominates who?) – exponentials or factorials?

That is, show that one of $f(n) = c^n$, $g(n) = n!$ dominates the other (for any $c \in \mathbb{R}_+$ say).

If you do more computer science, you will use Big O. You count (roughly, since constant multiples don't matter) the number of steps an algorithm takes on inputs of size $n \in \mathbb{N}$.

The algorithm is "fast" if the number of steps is $O(n)$ or maybe $O(n \log n)$, and "bad" if it takes $O(c^n)$ or worse steps.

Issues: is it bad if it takes this many steps on *all* inputs, or some inputs, or most inputs.

If you do more computer science, you will use Big O. You count (roughly, since constant multiples don't matter) the number of steps an algorithm takes on inputs of size $n \in \mathbb{N}$.

The algorithm is "fast" if the number of steps is $O(n)$ or maybe $O(n \log n)$, and "bad" if it takes $O(c^n)$ or worse steps.

Issues: is it bad if it takes this many steps on *all* inputs, or some inputs, or most inputs.

See Worksheet 6 for a table of standard functions and their names used in time complexity.

Exercise: prove that the worst case running time for the Euclidean algorithm is when the input is two of the Fibonacci numbers (worksheet 3)

Exercise: prove that the worst case running time for the Euclidean algorithm is when the input is two of the Fibonacci numbers (worksheet 3)

Standard exercise in comp sci: compare running times for algorithms which sort a list of numbers: bubble sort versus merge sort.

Recall from the Canvas "Welcome" page:
`https://www.youtube.com/watch?v=kVgy1GSDHG8`

And here is another video that mentions Big O:
`https://www.youtube.com/watch?v=xFFs9UgOAlE`

Next time:

- pigeonhole principle