Big-Oh notation

William Hendrix

Lecture 4

Today

- Review: Big-Oh

- Big-Omega, Big-Theta

- Big-Oh Properties

Big-Oh notation

- Describes *asymptotic growth rate* of functions
  - Ignores coefficients
  - Ignores behavior for small *n*

- Big-Oh: $O(f(n))$
  - Most common descriptor
  - Analogous to "at most" or "≤"
  - Good for worst-case analysis
  - Formal definition:

$f(n) = O(g(n))$ if and only if there exist positive constants c and n_0 such that $f(n) \leq cg(n)$ for all $n \geq n_0$.

  - $f(n) = O(g(n))$, not $f(n) \in O(g(n))$

Back to our example…

```
data:  an array of integers to find the min
n:     the number of values in data
```

Min algorithm:

```
min = 1
for i = 2 to n
   if data[i] < data[min]
      min = i
   end
end
return min
```

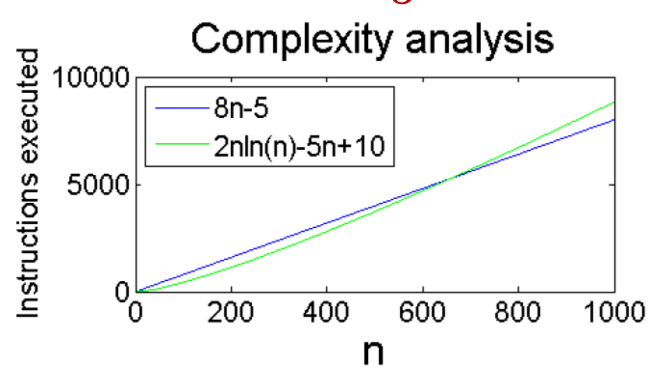
Ops per line   Times executed

…   …

Total ops: ≤ 8(n-1) + 3 = 8n - 5

Question: is this better or worse than an algorithm that takes at most $2n \ln n - 5n + 10$ ops?

Better unless n < 658



Worst case: *O(n)*

Don't need to count instructions!

Other algorithm: *O(n*lg*(n))*

Conclusion: Our algorithm is better for sufficiently large *n*.

Big-Omega

- Lower bound *("at least")*

$f(n) = \Omega(g(n))$ if and only if there exist positive constants c and n_0 such that $f(n) \geq cg(n)$ for all $n \geq n_0$.

- **Example**
  - Prove that $7n^2 + 19n - 4444 = \Omega(n)$.

Proof. If $n \geq 4444$,

$$\begin{aligned} 7n^2 + 19n - 4444 &\geq 19n - 4444 \\ &\geq 19n - n \\ &= 18n \end{aligned}$$

Therefore, there exist positive constants $c = 19$ and $n_0 = 4444$ such that $7n^2 + 19n - 4444 \geq cn^2$ for all $n \geq n_0$. □

Big-Theta

- Upper *and* lower bound *("same rate as")*

> $f(n) = \Theta(g(n))$ if and only if there exist positive constants c_1, c_2, and n_0 such that $c_1 g(n) \le f(n) \le c_2 g(n)$ for all $n \ge n_0$.

- **Example**
  - Prove that $7n^2 + 19n - 4444 = \Theta(n^2)$.

Proof. If $n \ge 4444$,

$$\begin{aligned} 7n^2 + 19n - 4444 &\ge 7n^2 + 19n - n \\ &= 7n^2 + 18n \\ &\ge 7n^2 \\ 7n^2 + 19n - 4444 &\le 7n^2 + 19n \\ &\le 7n^2 + n^2 \\ &= 8n^2 \end{aligned}$$

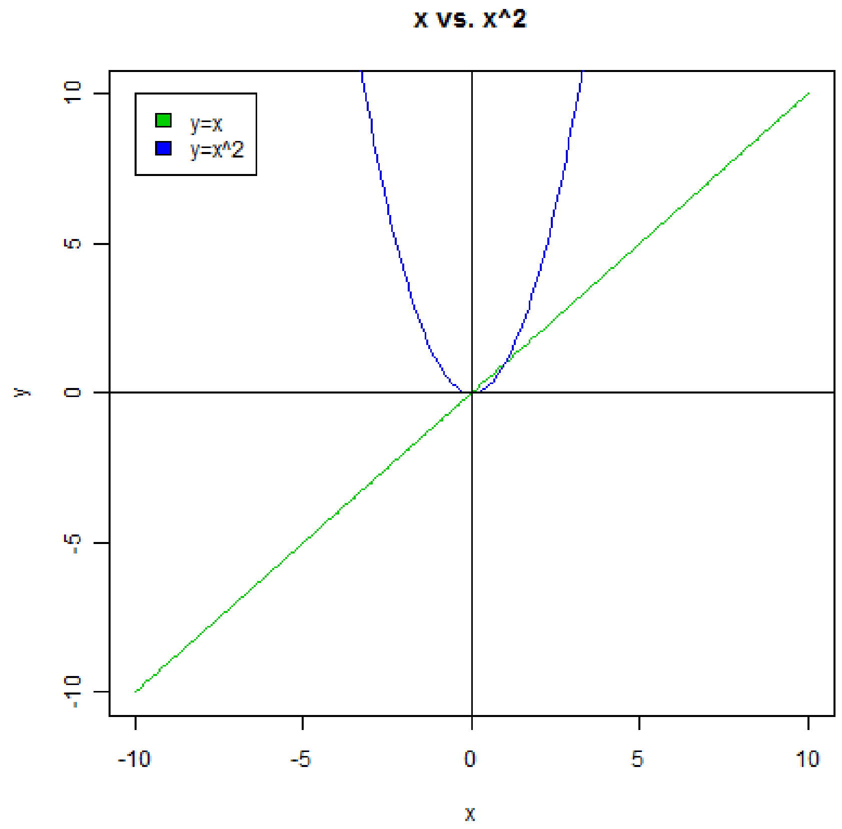
Therefore, there exist positive constants $c_1 = 7$, $c_2 = 8$, and $n_0 = 4444$ such that $c_1 n^2 \le 7n^2 + 19n - 4444 \le c_2 n^2$ for all $n \ge n_0$. $\square$

Proof advice

- General proof strategy: find values of c and n_o
- Finding smallest c or n_o isn't necessary
  - Choosing well can make your life easier, though
- Small powers of n are bounded above by larger powers
- Functions with negative coefficients are bounded above by zero
- Large coefficients are bounded above by n
  - With sufficiently large n_o

Big-Oh example

- Prove that $n = O(n^2)$.



$$\textit{Goal:}\ n \leq (\underline{\hspace{2cm}})n^2, \forall n \geq (\underline{\hspace{2cm}})$$

Big-Oh example

- Prove that $n = O(n^2)$.

Obviously true!

$1 \le n, \forall n \ge 1$ — Divide by n on both sides

$n \le n^2, \forall n \ge 1$

$n \le (1)n^2, \forall n \ge (1)$ — *Hypothesis:* $c = 1$, $n_0 = 1$

Goal: $n \le (\underline{\qquad})n^2, \forall n \ge (\underline{\qquad})$

Proof. Note that $1 \le n$ for all $n \ge 1$.

Since $n \ge 1$, we can multiply both sides by n: $n \le n^2$, for all $n \ge 1$.

Let $c = 1$ and $n_0 = 1$.
So, $n \le cn^2$, for all $n \ge n_0$.

Since there are positive constants c and n_0 such that $n \le cn^2$ for all $n \ge n_0$, $n = O(n^2)$. $\square$

Big-Oh exercises

- Use the *formal definitions* of Big-Oh, Big-Omega, and Big-Theta to prove the following:

1. $$\frac{n(n+1)}{2} = O(n^2)$$

2. If $f(n) = O(g(n))$, $g(n) = \Omega(f(n))$.

3. If $f(n) = \Omega(g(n)$ and $g(n) = \Omega(h(n))$, then $f(n) = \Omega(h(n))$.

Big-Oh exercises

- Use the *formal definitions* of Big-Oh, Big-Omega, and Big-Theta to prove the following:

1. *Proof.* If $n \geq 1$, $n \leq n^2$, so

$$\begin{aligned} \frac{n(n+1)}{2} &\leq n(n+1) \\ &= n^2 + n \\ &\leq n^2 + n^2 \\ &= 2n^2 \end{aligned}$$

Hence, there exist constants $c = 2$ and $n_0 = 1$ such that $\frac{n(n+1)}{2} \leq cn^2$ for all $n \geq n_0$, so $\frac{n(n+1)}{2} = O(n^2)$. $\square$

Big-Oh exercises

- Use the *formal definitions* of Big-Oh, Big-Omega, and Big-Theta to prove the following:

2. *Proof.* If $f(n) = O(g(n))$, there exist positive constants c_1 and n_0 such that $f(n) \leq c_1 g(n)$ for all $n \geq n_0$. Since $c_1 > 0$, we can multiply both sides of this expression by $\frac{1}{c}$, yielding $\left(\frac{1}{c}\right) f(n) \leq g(n)$ for all $n \geq n_0$. Thus, there exist positive constants $c_2 = \frac{1}{c}$ and $n_2 = n_0$ such that $g(n) \geq c_2 f(n)$ for all $n \geq n_2$, so $g(n) = \Omega(f(n))$. $\square$

3. *Proof.* If $f(n) = O(g(n))$ and $g(n) = O(h(n))$, there exist positive constants c_1, c_2, n_0, and n_1 such that $f(n) \leq c_1 g(n)$ for all $n \geq n_0$ and $g(n) \leq c_2 h(n)$ for all $n \geq n_1$. In particular, if we let $n_2 = \max\{n_0, n_1\}$, $f(n) \leq c_1 g(n)$ and $g(n) \leq c_2 h(n)$ for all $n \geq n_2$, so $f(n) \leq c_1(c_2 h(n))$. Thus, there exist constants $c_3 = c_1 c_2$ and $n_2 = \max\{n_0, n_1\}$ such that $f(n) \leq c_3 h(n)$ for all $n \geq n_2$, so $f(n) = O(h(n))$. $\square$

Observations on Big-Oh

- Big-Oh can be larger than needed
  - $n^3 = O(n^3), O(n^4), O(n^5)$ ...
- Big-Omega can be smaller than needed
- *Analogy*: Big-Oh "acts like" ≤, Big-Omega ≥, and Big-Theta =
- We will generally look for tight upper bounds ($O(f(n))$) in this class
- Most algorithms we discuss will belong to the following classes:

$$O(1) \ll O(\lg n) \ll O(n) \ll O(n \lg n) \ll O(n^2) \ll O(n^3) \ll O(2^n) \ll O(n!)$$

  - Constant, logarithmic, linear, n log n (or "linearithmic"), quadratic, cubic, exponential, or factorial

Connection to calculus

- You can also determine *O*, *Ω*, and *Θ* by limits:

g grows faster ⟶ $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \quad \rightarrow f(n) = O(g(n))$$

Same growth rate ⟶ $$\lim_{n\to\infty} \frac{f(n)}{g(n)} \in (0, \infty) \rightarrow f(n) = \Theta(g(n))$$

g grows slower ⟶ $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \quad \rightarrow f(n) = \Omega(g(n))$$

- Standard rules for taking limits apply
  – Including L'Hôpital's Rule

Coming up

- Big-Oh properties
- Big-Oh practice

- **Homework 3** is due tomorrow
- **Homework 4** is due Friday

- **Recommended readings:** Sections 2.3 and 2.4
- **Practice problems:** 2-7, 2-8, 2-10, 2-17, 2-18, 2-23 (pp. 58-60)