University of Colorado Boulder

Recitation: Asymptotic notation, time complexity of code

Basics

Let $f, g : \mathbb{N} \to \mathbb{R}$ be sequences of real numbers. We say that $f(n) \in O(g(n))$ if there is a constant c and a threshold natural number N such that for all $n \geq N$, we have

$$f(n) \leq c \cdot g(n)$$

Let $f, g : \mathbb{N} \to \mathbb{R}$ be sequences of real numbers. We say that $f(n) \in \Omega(g(n))$ if there is a constant c and a threshold natural number N such that for all $n \geq N$, we have

$$f(n) \geq c \cdot g(n)$$

Let $f, g : \mathbb{N} \to \mathbb{R}$ be sequences of real numbers. We say that $f(n) \in \Theta(g(n))$ if it holds that both $f \in O(g)$ and $f \in \Omega(g)$ such that we can find c_1, c_2, N which would satisfy for all $n \geq N\}$

$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$$

Question 1. Briefly justify that $\frac{1}{2}n^2 - 3n = \Theta\left(n^2\right)$.
Answer: To do so, we must determine constants c_1, c_2, and n_0 greater than zero such that

$$c_1 n^2 \leq \frac{1}{2}n^2 - 3n \leq c_2 n^2$$

for all $n \geq n_0$. Dividing by n^2 yields

$$c_1 \leq \frac{1}{2} - \frac{3}{n} \leq c_2$$

We can make the right-hand inequality hold for any value of $n \geq 1$ by choosing any constant $c_2 \geq 1/2$. Likewise, we can make the left-hand inequality hold for any value of $n \geq 7$ by choosing any constant $c_1 \leq 1/14$. Thus, by choosing $c_1 = 1/14$, $c_2 = 1/2$, and $n_0 = 7$, we can verify that $\frac{1}{2}n^2 - 3n = \Theta\left(n^2\right)$. This is not a unique triple of constants. Other choices for the constants exist, but the important thing is that some choice exists.

Question 2. Next verify that $6n^3 \neq \Theta\left(n^2\right)$.
Answer: Assume for the purposes of contradiction that c_2 and n_0 exist such that $6n^3 \leq c_2 n^2$ for all $n \geq n_0$. But then dividing by n^2 yields $n \leq c_2/6$, which cannot possibly hold for arbitrarily large n, since c_2 is constant.

Limit comparison test

If the limit $L := \lim_{x\to\infty} \frac{f(x)}{g(x)}$ exists, then:

1. If $0 < L < \infty$, then $f(n) \in \Theta(g(n))$.
2. If $L = 0$, then $f(n) \in O(g(n))$, but $f(n) \notin \Theta(g(n))$.
3. If $L = \infty$, then $g(n) \in O(f(n))$, but $g(n) \notin \Theta(f(n))$.

Also recall L'Hôpital's rule: suppose g and f are both differentiable functions, with either

- $\lim_{x\to\infty} f(x) = 0$ and $\lim_{x\to\infty} g(x) = 0$; or
- $\lim_{x\to\infty} f(x) = \pm\infty$ and $\lim_{x\to\infty} g(x) = \pm\infty$

If $\lim_{x\to\infty} \frac{f(x)}{g(x)}$ exists, then

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = \lim_{x\to\infty} \frac{f'(x)}{g'(x)}$$

.

Question 3. Order the two functions $f(n) = n^2 + 3n + 5$, and $g(n) = 3^n$.
Answer: The limit

$$\lim_{n\to\infty} \frac{n^2 + 3n + 5}{3^n}$$

has the indeterminate form $\frac{\infty}{\infty}$. Furthermore, $f(n) = n^2+3n+5$ and $g(n) = 3^n$ are both differentiable. In order to apply L'Hopital's rule, first establish that the following limit exists:

$$\lim_{n\to\infty} \frac{f'(n)}{g'(n)} = \lim_{n\to\infty} \frac{2n+3}{\ln(3)\cdot 3^n}.$$

$f'(n) = 2n + 3$ and $g'(n) = \ln(3)\cdot 3^n$ are differentiable functions, apply L'Hopital's rule:

$$\lim_{n\to\infty} \frac{f''(n)}{g''(n)} = \lim_{n\to\infty} \frac{2}{(\ln(3))^2\cdot 3^n} = 0.$$

$$\lim_{n\to\infty} \frac{2n+3}{\ln(3)\cdot 3^n} = \lim_{n\to\infty} \frac{2}{(\ln(3))^2\cdot 3^n} = 0.$$

$$\lim_{n\to\infty} \frac{n^2+3n+5}{3^n} = \lim_{n\to\infty} \frac{2n+3}{\ln(3)\cdot 3^n} = \lim_{n\to\infty} \frac{2}{(\ln(3))^2\cdot 3^n} = 0.$$

So by the limit comparison test, $n^2 + 3n + 5 \in O\left(3^n\right)$.

Question 4. (1) Show that (a) $n^2 + 1 = \Omega(n)$, (b) $n^2 + 3n + 4 = \Theta\left(n^2\right)$.
(2) Order the two functions $n^2 \log(n)$ and $n(\log(n))^{10}$.
(3) Show that $\log_{\log 5}((\log n)^{100}) = \Theta(\log\log n)$.

Answer: (1) (a) $n^2+1 = \Omega(n) \Leftrightarrow \lim_{n\to\infty} \frac{n^2+1}{n} = \infty$, (b) $n^2+3n+4 = \Theta\left(n^2\right) \Leftrightarrow \lim_{n\to\infty} \frac{n^2+3n+4}{n^2} = \lim_{n\to\infty} \left(1 + \frac{3}{n} + \frac{4}{n^2}\right) = 1$.

(2) Compare n to $\log(n)^k$. Notice that

$$\lim_{n\to\infty} \frac{n}{\log(n)^k} = \lim_{n\to\infty} \frac{1}{k\log(n)^{k-1}\frac{1}{n}} = \lim_{n\to\infty} \frac{n}{k\log(n)^{k-1}}$$

Applying L'Hopital's rule k times, we find out that the limit diverges, for any constant $k > 1$. Thus we find that, for all $k > 1, n\log(n)^k = O\left(n^2\log(n)\right)$.

(3) $\log_{\log 5}((\log n)^{100}) = \Theta(\log_{\log 5}((\log n)) = \Theta(\log\log n)$. (Why?)

Question 5. Analyze the time complexity:

```
procedure Search(Array A[1, . . . , n], key)
    found ← false
    for i ← 1;i | len(A);i ← i + 1 do
        if A[i] == key then
        found ← true
  return found
```

- Line 2, single assignment statement. So Line 2 takes 1 step.
- Initializing the loop at Line 3 takes a single assignment, so 1 step.
- The loop itself takes n iterations. At each iteration we have:
  - The comparison that $i \leq \text{len}(A)$, which takes 1 step.
  - The variable update $i \leftarrow i + 1$, which takes 2 steps: one step for the $i + 1$ computation, and one step for the assignment.
  - In the body of the loop, we have the comparison at Line 4, which takes 2 steps: one for the array access and one for the comparison.
  - At Line 5, we have a single assignment, which takes 1 step.

Thus, the loop starting at Line 3 has time complexity:

$$\sum_{i=1}^{n}(1 + 2 + 2 + 1) = \sum_{i=1}^{n} 6 = 6n.$$

- We have the return statement after Line 5, which takes 1 step. As we have an iterative algorithm, we add up the costs to obtain:

$$\begin{aligned} T(n) &= 1 + 1 + 6n + 1 \\ &= 6n + 3. \end{aligned}$$

Thus, $T(n) \in \Theta(n)$.

Recitation: Asymptotic notation, time complexity of code

Basics

Let $f, g : \mathbb{N} \to \mathbb{R}$ be sequences of real numbers. We say that $f(n) \in O(g(n))$ if there is a constant c and a threshold natural number N such that for all $n \geq N$, we have

$$f(n) \leq c \cdot g(n)$$

Let $f, g : \mathbb{N} \to \mathbb{R}$ be sequences of real numbers. We say that $f(n) \in \Omega(g(n))$ if there is a constant c and a threshold natural number N such that for all $n \geq N$, we have

$$f(n) \geq c \cdot g(n)$$

Let $f, g : \mathbb{N} \to \mathbb{R}$ be sequences of real numbers. We say that $f(n) \in \Theta(g(n))$ if it holds that both $f \in O(g)$ and $f \in \Omega(g)$ such that we can find c_1, c_2, N which would satisfy for all $n \geq N\}$

$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$$

Question 1. Briefly justify that $\frac{1}{2}n^2 - 3n = \Theta\left(n^2\right)$.

Question 2. Next verify that $6n^3 \neq \Theta\left(n^2\right)$.

Limit comparison test

If the limit $L := \lim_{x\to\infty} \frac{f(x)}{g(x)}$ exists, then:

1. If $0 < L < \infty$, then $f(n) \in \Theta(g(n))$.
2. If $L = 0$, then $f(n) \in O(g(n))$, but $f(n) \notin \Theta(g(n))$.
3. If $L = \infty$, then $g(n) \in O(f(n))$, but $g(n) \notin \Theta(f(n))$.

Also recall L'Hôpital's rule: suppose g and f are both differentiable functions, with either

- $\lim_{x\to\infty} f(x) = 0$ and $\lim_{x\to\infty} g(x) = 0$; or
- $\lim_{x\to\infty} f(x) = \pm\infty$ and $\lim_{x\to\infty} g(x) = \pm\infty$

If $\lim_{x\to\infty} \frac{f(x)}{g(x)}$ exists, then

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = \lim_{x\to\infty} \frac{f'(x)}{g'(x)}$$

Question 3. Order the two functions $f(n) = n^2 + 3n + 5$, and $g(n) = 3^n$.

Question 4. (1) Show that (a) $n^2+1=\Omega(n)$, (b) $n^2+3n+4=\Theta\left(n^2\right)$.
(2) Order the two functions $n^2\log(n)$ and $n(\log(n))^{10}$.
(3) Show that $\log_{\log 5}((\log n)^{100})=\Theta(\log\log n)$.

Question 5. Analyze the time complexity:

```
procedure Search(Array A[1, . . . , n], key)
    found ← false
    for i ← 1;i  len(A);i ← i + 1 do
        if A[i] == key then
        found ← true
  return found
```

```
procedure foo_d(integer n):
  for i = 1, i <= n
      i = i + 2
      print ‘outer’
      for j = 1, j <= i
          j = j + 3
          print ‘inner’
```

```
procedure foo_d(integer n):
  for i = 1, i <= n
      i = i + 2
      print ‘outer’
      for j = 1, j <= i
          j = j + 3
          print ‘inner’
```

```
procedure foo_i(integer n):
  for i = 1, i <= n
      i = 2 * i
      print ‘outer’
      for j = 1, j <= n
          j = 3 * j
          print ‘inner’
```

logarithm computation rules.

1. $\log_b(mn) = \log_b(m) + \log_b(n)$

2. $\log_b\left(\frac{m}{n}\right) = \log_b(m) - \log_b(n)$

3. $\log_b(n^P) = P\log_b n$

4. $\log_b(n) = \dfrac{\log_p(n)}{\log_p(b)}$

5. $\log_b(n) = \log_b(P) \cdot \log_p n$

6. $a^{\log_c b} = b^{\log_c a}$

we can prove rule 6:

Assume $a = x^n$, $b = x^m$

$$a^{\log b} = (x^n)^{\log x^m}$$
$$= (x^n)^{m \log x}$$
$$= x^{n \cdot m \log x}$$
$$= (x^m)^{n \log x}$$
$$= (x^m)^{\log x^n}$$
$$= b^{\log a}.$$

Differential rules.

1. $$\frac{d}{dx}\log_a x = \frac{1}{x\ln(a)}$$

2. $$\frac{d}{dx}a^x = a^x \cdot \ln(a).$$