Algorithm Design Manual
Skiena - Chapter 2
Notes

Melissa Auclaire

Formal Definition of Little Oh
$f(n)$ is $o(g(n))$ if $f(n) = O(g(n))$ and $f(n) \neq \Theta(g(n))$.

The formal definitions associated with Big Oh notation are as follows:

- $f(n) = O(g(n))$ means $c \cdot g(n)$ is an upper bound on $f(n)$. Thus there exists some constant c such that $f(n)$ is always $\leq c \cdot g(n)$, for large enough n (i.e., $n \geq n_0$ for some constant n_0).
- $f(n) = \Omega(g(n))$ means $c \cdot g(n)$ is a lower bound on $f(n)$. Thus there exists some constant c such that $f(n)$ is always $\geq c \cdot g(n)$, for all $n \geq n_0$.
- $f(n) = \Theta(g(n))$ means $c_1 \cdot g(n)$ is an upper bound on $f(n)$ and $c_2 \cdot g(n)$ is a lower bound on $f(n)$, for all $n \geq n_0$. Thus there exist constants c_1 and c_2 such that $f(n) \leq c_1 \cdot g(n)$ and $f(n) \geq c_2 \cdot g(n)$. This means that $g(n)$ provides a nice tight bound on $f(n)$.

— $\underbrace{f(n) = O(g(n))}_{g \text{ dominates } f} \rightarrow g \gg f$

$\underbrace{g(n) = O(f(n))}_{f \gg g}$ if $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$

$f(n)$ dominates $g(n)$ if $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$

$$\lim_{n\to\infty} \frac{g(n)}{f(n)} = \lim_{n\to\infty} \frac{n^2}{2n^2} = \lim_{n\to\infty} \frac{1}{2} \neq 0$$

Almost all polynomial comparisons, generally the one with the greatest exponential term dominates because

$$\lim_{n\to\infty} \frac{n^b}{n^a} = \lim_{n\to\infty} n^{b-a} = 0$$ e.g. $n^{1.2}$ dominates $n^{1.199}$

$f \gg g$ $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$	$g \gg f$ $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$
$f \gg g$ $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$	$g \gg f$ $\lim_{n\to\infty} \frac{g(n)}{f(n)} = \infty$

Texbook:

$f(n)$ dominates $g(n)$ if $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$

g dominates f when $f(n) = O(g(n)) \Rightarrow g \gg f$

$\therefore f$ dominates g when $g(n) = O(f(n)) \Rightarrow f \gg g$

$g(n)$ dominates $f(n)$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

$f(n) = O(g(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

1. $g(n) = O(f(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$
2. $f(n) = \Omega(g(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$
3. $f(n) = O(g(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$
4. $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \Rightarrow g(n) = O(f(n)) \Rightarrow f(n) = \Omega(g(n))$

Limits and O, Ω, Θ

$f(n) = O(g(n)) \Leftarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

$f(n) = \Omega(g(n)) \Leftarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$

$f(n) = \Theta(g(n)) \Leftarrow$

$0 < \lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty$

Efficiency classes (order):

$n! \gg c^n \gg n^3 \gg n^2 \gg n^{1+\epsilon} \gg n\log n \gg n \gg \sqrt{n} \gg \log^2 n \gg \log n \gg \log n / \log\log n \gg \alpha(n) \gg 1$

Melissa Auclaire

Algoritm Design Manual
Skiena - Chapter 2
Exercises

2-1 [3] What value is returned by the following function? Express your answer as a function of n. Give the the worst-case running time in big Oh notation.

```
function mystery(n)
    r := 0
    for i := 1 to n-1 do
        for j := i+1 to n do
            for k := 1 to j do
                r := r + 1
    return(r)
```

Big O: $O(n^3)$

when n = 0

```
for i := 1 to -1 do
    for j := i + 1 to 0 do
        for k := 1 to j do
            r := r + 1
```

mystery(0) => 0
mystery(1) => 0

Evaluate n = 2

```
for i := 1 to 1 do
    for j := i+1 to 2 do
        for k := 1 to j do
            r := r + 1
```

mystery(2) => 2

Evaluate Big O:

Is $\frac{n^3+n^2}{2} - \frac{2n^3+3n^2+n}{12} - \frac{n^2-n}{4}$ $O(n^3)$?

Yes iff there exists a constant $c > 0$ such that for all sufficiently large n $f(n) \le c \cdot g(n)$.

$\frac{n^3}{2} - \frac{n^3}{6} - \frac{n^2}{4}$

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k=1}^{j} 1 =$$

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} j =$$

$$\sum_{i=1}^{n-1} \left(\sum_{j=1}^{n} j - \sum_{j=1}^{i} j \right) =$$

$$\sum_{i=1}^{n-1} \left(\frac{n(n+1)}{2} - \frac{i(i+1)}{2} \right) =$$

$$\frac{1}{2} \sum_{i=1}^{n-1} n^2 + n - i^2 - i =$$

$$\frac{1}{2} \left((n-1)n^2 + (n-1)n - \left(\frac{n(n+1)(2n+1)}{6} - n^2 \right) - \left(\frac{n(n+1)}{2} - n \right) \right) =$$

$$f(n) = \frac{n(n(n+1))}{2} - \frac{n(n+1)(2n+1)}{12} - \frac{n(n+1)}{4}$$

$$f(n) = \frac{n(n^2+n))}{2} - \frac{(n^2+n)(2n+1)}{12} - \frac{n^2+n}{4}$$

$$f(n) = \frac{n^3+n^2}{2} - \frac{2n^3+3n^2+n}{12} - \frac{n^2-n}{4}$$

Melissa Auclaire

Algorithm Design Manual
Skiena - Chapter 2
Exercises

2-2 [3] What value is returned by the following function? Express your answer as a function of n. Give the worst case running time using Big Oh notation.

```
function pesky(n)
    r := 0
    for i := 1 to n do
        for j := 1 to i do
            for k := j to i+j do
                r := r + 1
```

```
function pesky(0)  => 0
    r := 0
    for i := 1 to 0 do
```

```
function pesky(1) => 2
    r := 0
    for i := 1 to 1 do
        for j := 1 to 1 do
            for k := 1 to 2 do
                r := r + 1
```

Big O : $O(n^3)$

Evaluate Big O

$$\frac{n(n+1)(n+2)}{3} = \frac{(n^2+n)(n+2)}{3} =$$

$$\frac{n^3+3n^2+2n}{3}$$

$$\frac{n^3}{3} + \frac{3n^{\cancel{2}}}{\cancel{3}} + \frac{2\cancel{n}}{\cancel{3}}$$

Big $O(n^3/3)$

$\therefore O(n^3)$

$$\sum_{i=1}^{n}\sum_{j=1}^{i}\sum_{K=j}^{i+j} 1 = \sum_{i=1}^{n}\sum_{j=1}^{i} i+j-j+1 = \sum_{i=1}^{n}\left(\sum_{j=1}^{i} i + \sum_{j=1}^{i} 1\right) =$$

$$\sum_{i=1}^{n}\left((i+1)\sum_{j=1}^{i} 1\right) = \sum_{i=1}^{n}(i+1)\,i = \sum_{i=1}^{n} i^2 + i = \sum_{i=1}^{n} i^2 + \sum_{i=1}^{n} i =$$

$$\frac{n(n+1)(2n+1)}{6} + \frac{(n-1+1)(n+1)}{2} =$$

$$\frac{n(n+1)(2n+1)}{6} + \frac{n(n+1)}{2} =$$

$$\frac{n(n+1)(2n+1)+3n(n+1)}{6} = \frac{n(n+1)(2n+4)}{6} = \frac{2n(n+1)(n+2)}{6} = \frac{n(n+1)(n+2)}{3}$$

2-3 [5] What value is returned by the following function? Express your answer as a function of n. Give the worst case running time using Big Oh notation.

```
function prestiferous(n)
    r := 0
    for i := 1 to n do
        for j := 1 to i do
            for K := j to i + j do
                for l := 1 to i+j-k do
                    r := r + 1
```

```
prestiferous(1) => 1
    r := 0
    for i := 1 to 1 do
        for j := 1 to 1 do
            for k := 1 to 2 do
                for l := 1 to 2-K do
                    r := r+1
```

```
prestiferous(0)  => 0
    r := 0
    for i := 1 to 0 do
```

Melissa Auclaire

2-3 [5] contd

$$\sum_{k=j}^{i+j}(j+i-k) = \sum_{k=1}^{i} k = \frac{i(i+1)}{2}$$

Big O (n^4)

Formula

$$\sum_{i=1}^{n}\sum_{j=1}^{i}\sum_{k=j}^{i+j}\sum_{l=1}^{i+j-k} 1 =$$

$$\sum_{i=1}^{n}\sum_{j=1}^{i}\sum_{k=j}^{i+j}(j+i-k) =$$

$$\sum_{i=1}^{n}\sum_{j=1}^{i}\left(\frac{i(i+1)}{2}\right) =$$

$$\sum_{i=1}^{n}\left(\frac{i^2(i+1)}{2}\right) = \sum_{i=1}^{n}\left(\frac{(i^3+i^2)}{2}\right) =$$

$$\frac{1}{2}\left(\sum_{i=1}^{n} i^3 + \sum_{i=1}^{n} i^2\right) =$$

$$\frac{1}{2}\left(\frac{n^2(n+1)^2}{4} + \frac{n(n+1)(2n+1)}{6}\right) =$$

$$\frac{n^2(n+1)^2}{8} + \frac{(n^2+n)(2n+1)}{12} =$$

Derive Big O :

1st term $\frac{n^2(n+1)^2}{8} = \frac{(n^3+n^2)(n+1)}{8} = \frac{n^4+2n^3+n^2}{8}$

2nd term $\frac{(n^2+n)(2n+1)}{12} = \cancel{\frac{2n^3+3n^2+n}{12}}$

2-4 [8] What value is returned by the following function? Express your answer as a function of n. Give the worst-case running time using Big Oh notation.

```
function conundrum(n)
    r := 0
    for i := 1 to n do
        for j := i + 1 to n do
            for k := i + j - 1 to n do
                r := r + 1
```

Big O (n^2)

```
conundrum(0) => 0
r := 0
for i := 1 to 0 do

conundrum(1) => 0
r := 0
for i := 1 to 1 do
  for j := 1 + 1 to 1 do
```

Formula:

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}\sum_{k=i+j-1}^{n} 1 =$$

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}(-i-j+n+2) =$$

$$\sum_{i=1}^{n}\left(\frac{1}{2}(i-n)(3i-n-3)\right) =$$

$$\frac{1}{2}(n-1)n$$

Big O Derivation:

$$\frac{1}{2}(n-1)n = \frac{n(n-1)}{2} = \frac{n^2-n}{2} = \frac{n^2}{2} - \frac{n}{2}$$

Algorithm Design manual
Skiena - Chapter 2
Exercises

Melissa Auclaire

2-5 [5] Suppose the following algorithm is used to evaluate the polynomial

$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$

```
p := a0 ;
xpower := 1 ;
for i := 1 to n do
    xpower := x * xpower;
    p := p + ai * xpower
end
```

(a) How many multiplications are done in the worst case? How many additions?
(b) How many multiplications are done on the average?
(c) Can you improve this algorithm?

(a) $2n$ multiplications, n additions
(b) $2n$ multiplications

(c) Horner's method of synthetic division is faster than the given algorithm.

2-6 [3] Prove that the following algorithm for computing the maximum value in an array A[1...n] is correct.

```
function max(A)
    m := A[1]
    for i := 2 to n do
        if A[i] > m then m := A[i]
    return m
```

Base case: max ([0]) => m (n=1)

In max (1...n) for any n for a given n

Inductive step:

max (1...n+1)

if n+1 = max
    n+1 is returned as m

if n+1 ≠ max
    max (1...n+1) := max (1...n) as shown above ∎

Big O

2-7 [3] True or false?
(a) Is $2^{n+1} = O(2^n)$?
(b) Is $2^{2n} = O(2^n)$?

(a) $f(n) = O(g(n))$ iff

$\exists c \mid \forall n \; f(n) \le c \cdot g(n)$

$2^{n+1} = 2 \cdot 2^n \le c \cdot 2^n$ for any $c \ge 2$.

$\therefore$ true

(b) $f(n) = O(g(n))$ iff $\exists c \mid \forall n \; f(n) \le c \cdot g(n)$

$2^{2n} = (2^n)^2 = (2 \cdot 2^{n-1})^2 \le 2 \cdot 2^n$ for any $c \ge 2$.

$\therefore$ true

Algorithm Design Manual
Skiena - Chapter 2
Exercises

Melissa Auclaire

2-8 [5] For each of the following pairs of functions, either $f(n)$ is in $O(g(n))$, $f(n)$ is in $\Omega(g(n))$, or $f(n) = \Theta(g(n))$. Determine which relationship is correct and explain why.

(a) $f(n) = \log n^2$; $g(n) = \log n + 5$
(b) $f(n) = \sqrt{n}$; $g(n) = \log n^2$
(c) $f(n) = \log^2 n$; $g(n) = \log n$
(d) $f(n) = n$; $g(n) = \log^2 n$
(e) $f(n) = n\log n + n$; $g(n) = \log(n)$
(f) $f(n) = 10$; $g(n) = \log 10$
(g) $f(n) = 2^n$; $g(n) = 10n^2$
(h) $f(n) = 2^n$; $g(n) = 3^n$

a) $\log n^2 = 2\log n$
$2\log n \le 2\log n + 10$
$\log n^2 \le 2(\log n + 5)$
$\log n^2 \le c(\log n + 5)$ where $c = 2$
$\log n^2 = O(\log n + 5)$

$\log n + 5 \le \log n + 5\log n$
$\log n + 5 \le 6\log n$
$\log n + 5 \le 3(2)\log(n)$
$3\log n^2 \ge \log n + 5$
$\log n^2 \ge c(\log n + 5)$ where $c = \frac{1}{3}$
$\log n^2 = \Omega(\log n + 5)$
$\therefore \log n^2 = \Theta(\log n + 5)$

b) $\log n^2 = 2\log n$ ($g(n) = \log n^2 = 2\log n$)

$\lim_{n\to\infty} \frac{\sqrt{n}}{2\log n} = 2\lim_{n\to\infty} \frac{\sqrt{n}}{\log n} = \infty$

$g(n) = O(f(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$

i.e.
$f(n) = \Omega(g(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$
$f(n) = O(g(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

$\therefore f(n) = \Omega(g(n))$

c) $\lim_{n\to\infty} \frac{\log^2 n}{\log n} = \lim_{n\to\infty} \log(n) = \infty$ (see table in notes)

$\therefore \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \Rightarrow f(n) = \Omega(g(n))$ i.e. $g(n) = O(f(n))$

d) $\lim_{n\to\infty} \frac{n}{\log^2 n} = \lim_{n\to\infty} \left(\left(\frac{\sqrt{n}}{\log n}\right)^2\right) = \left(\lim_{n\to\infty} \frac{\sqrt{n}}{\log n}\right)^2 = \infty$

$\therefore \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \Rightarrow f(n) = \Omega(g(n))$

e) $\lim_{n\to\infty} \frac{n\log n + n}{\log n} = \lim_{n\to\infty} \left(\frac{n\log n}{\log n} + \frac{n}{\log n}\right) = \lim_{n\to\infty} \left(n + \frac{n}{\log n}\right) = \infty$

$\therefore \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \Rightarrow f(n) = \Omega(g(n))$

f) Both are $O(1)$.
$\therefore f(n) = \Theta(g(n))$

g) $\lim_{n\to\infty} \frac{2^n}{10n^2} = \frac{1}{10}\left(\lim_{n\to\infty} \frac{2^n}{n^2}\right) = \frac{1}{10}\left(\lim_{n\to\infty} \frac{(\ln 2)2^n}{2n}\right) =$
$\frac{1}{10}\left(\lim_{n\to\infty} \frac{(2\ln 2)2^n}{2}\right) = \frac{\ln 2}{10}\left(\lim_{n\to\infty} 2^n\right) =$
$\frac{\ln 2}{10} \lim_{n\to\infty} 2^n = \infty$

$\therefore \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \Rightarrow f(n) = \Omega(g(n))$

h) $\lim_{n\to\infty} \frac{2^n}{3^n} = \lim_{n\to\infty} \left(\frac{2}{3}\right)^n = 0$

$\therefore \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \Rightarrow f(n) = O(g(n))$

Algorithm Design Manual
Skiena - Chapter 2
Exercises

Melissa Auclaire

2-9 [3] For each of the following pairs of functions $f(n)$ and $g(n)$, determine whether $f(n) = O(g(n))$, $g(n) = O(f(n))$, or both.

a) $f(n) = \frac{n^2-n}{2}$, $g(n) = 6n$
b) $f(n) = n + 2\sqrt{n}$, $g(n) = n^2$
c) $f(n) = n\log n$, $g(n) = \frac{n\sqrt{n}}{2}$
d) $f(n) = n + \log n$, $g(n) = \sqrt{n}$
e) $f(n) = 2(\log n)^2$, $g(n) = \log n + 1$
f) $f(n) = 4n\log n + n$, $g(n) = \frac{(n^2-n)}{2}$

a) $\lim_{n\to\infty} \frac{n^2-n}{2} = \infty$, $\lim_{n\to\infty} 6n = \infty$

$\frac{n^2-n}{2} > 6n$ as $n \to \infty$

$\therefore g(n) = O(f(n))$

b) $\lim_{n\to\infty} n + 2\sqrt{n} = \infty$, $\lim_{n\to\infty} n^2 = \infty$

$n^2 > n + 2\sqrt{n}$ as $n \to \infty$

$\therefore f(n) = O(g(n))$

c) $\lim_{n\to\infty} n\log(n) = \infty$, $\lim_{n\to\infty} \frac{n\sqrt{n}}{2} = \infty$

$\lim_{n\to\infty} \frac{\log n}{\sqrt{n}} = 0$

$\therefore \frac{n\sqrt{n}}{2} > n\log n$ as $n \to \infty$

$\therefore f(n) = O(g(n))$

d) $\lim_{n\to\infty} n + \log n = \infty$, $\lim_{n\to\infty} \sqrt{n} = \infty$

$\lim_{n\to\infty} \frac{n}{\sqrt{n}} = \sqrt{\lim_{n\to\infty} n} = \sqrt{\infty} = \infty$

$\therefore n + \log n > \sqrt{n}$ as $n \to \infty$

$\therefore g(n) = O(f(n))$

e) $\lim_{n\to\infty} 2(\log n)^2 = \infty$, $\lim_{n\to\infty} \log n + 1 = \infty$

$\lim_{n\to\infty} \frac{(\log n)^2}{\log n} = \infty$

$\therefore (\log n)^2 > \log n$ as $n \to \infty$

$\therefore g(n) = O(f(n))$

f) $\lim_{n\to\infty} 4n\log n + n = \infty$, $\lim_{n\to\infty} \frac{n^2-n}{2} = \infty$

$\lim_{n\to\infty} \frac{4n\log n}{\frac{n^2}{2}} = \lim_{n\to\infty} \frac{n\log n}{n^2} = 0$

$\therefore \frac{n^2-n}{2} > 4n\log n + n$ as $n \to \infty$

$\therefore f(n) = O(g(n))$

2-10 [3] Prove that $n^3 - 3n^2 - n + 1 = \Theta(n^3)$.

$\lim_{n\to\infty} \frac{n^3-3n^2-n+1}{n^3} = \lim_{n\to\infty} \frac{1 - \frac{3}{n} - \frac{1}{n^2} + \frac{1}{n^3}}{1} = 1$

$\therefore f(n) = O(g(n))$ and $g(n) = O(f(n))$

$\therefore f(n) = \Theta(g(n))$. ∎

2-11 [3] Prove that $n^2 = O(2^n)$.

$\lim_{n\to\infty} \frac{n^2}{2^n} = \lim_{n\to\infty} 2^{-n} n^2 = 0$

$\therefore \lim_{n\to\infty} \frac{f(n) = n^2}{g(n) = 2^n} = \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \Rightarrow f(n) = O(g(n))$ ∎

2-12 [3] For each of the following pairs of functions, $f(n)$ and $g(n)$, give an appropriate positive constant $c \mid f(n) \le c \cdot g(n)$ for all $n > 1$.

(a) $f(n) = n^2 + n + 1$, $g(n) = 2n^3$
(b) $f(n) = n\sqrt{n} + n^2$, $g(n) = n^2$
(c) $f(n) = n^2 - n + 1$, $g(n) = \frac{n^2}{2}$

a) $1^2 + 1 + 1 \overset{?}{\le} 2(1)^3$

$3 \not\le 2$

$2^2 + 2 + 1 \overset{?}{\le} 2(2)^3$

$7 \le 16$

$\therefore c = 1$

c) $1^2 - 1 + 1 \overset{?}{\le} \frac{1^2}{2}$

$1 \not\le \frac{1}{2}$

$2^2 - 2 + 1 \overset{?}{\le} \frac{2^2}{2}$

$1 \le 2$

$\therefore c = 1$

b) $1\sqrt{1} + 1^2 \overset{?}{\le} 1^2$

$\sqrt{1} + 1 \not\le 1$

$2\sqrt{2} + 2^2 \overset{?}{\le} 2^2$

$2\sqrt{2} + 4 \not\le 4$

$2\sqrt{2} + 4 \overset{?}{\le} (3)4$

$2\sqrt{2} + 4 \le 12$

$\therefore c = 3$

Algorithm Design Manual
Skiena - Chapter 2
Exercises

2-13 [3] Prove that if $f_1(n) = O(g_1(n))$ and $f_2(n) = O(g_2(n))$, then $f_1(n) + f_2(n) = O(g_1(n) + g_2(n))$.

$a \le b,\ c \le d \Rightarrow a + c \le b + d$

$f_1(n) \le c \cdot g_1(n),$
$f_2(n) \le c \cdot g_2(n),$
$f_1(n) + f_2(n) \le c(g_1(n) + g_2(n))$
$\therefore f_1(n) + f_2(n) = O(g_1(n) + g_2(n))$ ∎

2-14 [3] Prove that if $f_1(N) = \Omega(g_1(n))$ and $f_2(n) = \Omega(g_2(n))$, then $f_1(n) + f_2(n) = \Omega(g_1(n) + g_2(n))$.

$c \cdot f_1(n) \ge g_1(n),$ $\quad a \ge b,\ c \ge d \Rightarrow a + c \ge b + d$
$c \cdot f_2(n) \ge g_2(n),$
$c(f_1(n) + f_2(n)) \ge g_1(n) + g_2(n)$
$\therefore f_1(n) + f_2(n) = \Omega(g_1(n) + g_2(n))$ ∎

2-15 [3] Prove that if $f_1(n) = O(g_1(n))$ and $f_2(n) = O(g_2(n))$, then $f_1(n) \cdot f_2(n) = O(g_1(n) \cdot g_2(n))$.

$a \le b,\ c \le d = ac \le bd$
$f_1(n) \le c \cdot g_1(n),$
$f_2(n) \le c \cdot g_2(n),$
$f_1(n) \cdot f_2(n) \le c(g_1(n) \cdot g_2(n))$
$\therefore f_1(n) \cdot f_2(n) = O(g_1(n) \cdot g_2(n))$ ∎

2-16 [5] Prove that for all $k \ge 1$ and all sets of constants $\{a_k, a_{k-1}, \dots, a_1, a_0\} \in R$
$a_k n^k + a_{k-1} n^{k-1} + \dots + a_1 n + a_0 = O(n^k)$

Base case $k = 1$
$a_1 n + a_0 = O(n^k)$
$a_1 n + a_0 = O(n)$

$a_1 n + a_0 \le c \cdot n$
$1n + 0 \le c \cdot n$
$n \le c \cdot n$ when $c \ge 1$
$\therefore a_1 n + a_0 = O(n^k)$

Inductive Step: $k \to k+1$
let $k = 1$
$a_{k+1} n^k + a_k n^k + \dots + a_1 n + a_0 = O(n^{k+1})$
$\exists c \mid$ $a_2 n^2 + a_1 n + a_0 \le c \cdot n^2$
$2n + n \le c \cdot n^2$ when $n \ge 2,\ c = 2$
$6 \le 8$
let $k = 2$
$a_{k+1} n^{k+1} + a_k n^k + \dots + a_1 n + a_0 = O(n^{k+1})$
let $k = 2$

$a_3 n^3 + a_2 n^2 + a_1 n + a_0 = O(n^3)$
$3n^3 + 2n^2 + 1n + 0 \le c \cdot n^3$
$3n^3 + 2n^2 + 1n + 0 \le c n^3$

Algorithm Design Manual
Skiena - Chapter 2
Exercises

Melissa Auclaire

2-16 [5] contd

$3n^3 + 2n^2 + n \le 5n^3$ where $c = 5$, $n = 2$

$3(8) + 2(4) + 2 \le 5(8)$

$34 \le 40 \qquad 40 - 34 = 6$

$3n^3 + 2n^2 + n \le 5n^3$ where $c = 5$, $n = 3$

$3(27) + 2(9) + 3 \le 5(27)$

$81 + 18 + 3 \le 135$

$102 \le 135 \qquad 135 - 102 = 33$

$\therefore a_{k+1}n^{k+1} + a_k n^k + \ldots + a_1 n + a_0 \le c \cdot n^{k+1}$ where $c = 5$

$\therefore a_k n^k + \ldots + a_1 n + a_0 = O(n^k)$ ■

2-17 [5] Show that for any real constants a and b, $b > 0$

$(n+a)^b = \Theta(n^b)$

Base case : ($a = 0$, $b = 1$)

$(n+a)^b = \Theta(n^b)$

$n^b = \Theta(n^b)$

$\lim_{n \to \infty} \frac{n^b}{n^b} = 1 \qquad \therefore 0 < \lim_{n \to \infty} \frac{f(n)}{g(n)} < \infty \Rightarrow$

$\therefore \lim \frac{f(n)}{g(n)} = 1 \qquad f(n) = \Theta(g(n)) \qquad n^b = \Theta(n^b)$

e.x. $f(n) \le c_1 \cdot g(n) \qquad c_1 = 1, c_2 = 1, n \ge 0$

$g(n) \ge c_2 \cdot g(n) \qquad f(n) = g(n)$

2-18 [5] List the functions below from the lowest to the highest order. If any two or more are of the same order, indicate which.

$\cancel{n}$. $\cancel{2^n}$. $n\cancel{\lg n}$. $\cancel{\ln n}$.

$\cancel{n - n^3 + 7n^5}$. $\cancel{\lg n}$. $\cancel{\sqrt{n}}$. $\cancel{e^n}$.

$\cancel{n^2 + \lg n}$. $\cancel{n^2}$. $\cancel{2^{n-1}}$. $\lg\cancel{\lg n}$.

$\cancel{n^3}$. $(\cancel{\lg n})^2$ $\cancel{n!}$. $\cancel{n^{1+\varepsilon}}$ where $0 < \varepsilon < 1$.

$n! \gg e^n / 2^n / 2^{n-1} \gg n - n^3 + 7n^5 \gg n^3 \gg n^2 + \lg n / n^2 \gg n^{1+\varepsilon} \gg$
$n \lg n \gg n \gg \sqrt{n} \gg (\lg n)^2 \gg \log n / \ln n \gg \lg \lg n$

Algorithm Design Manual
Skiena - Chapter 2
Exercises

Melissa Auclaire

2-19 [5] List the functions below from the lowest to the highest order. If any two or more are of the same order, indicate which.

$\sqrt{n}$	$n!$	2^n
$n \log n$	$n - n^3 + 7n^5$	$n^2 + \log n$
n^2	n^3	$\log n$
$n^{\frac{1}{3}} + \log n$	$(\log n)^2$	$n!$
$\ln n$	$\frac{n}{\log n}$	$\log \log n$
$(\frac{1}{3})^n$	$(\frac{3}{2})^n$	6

$n! >> (\frac{3}{2})^n \,/\, 2^n >> n - n^3 + 7n^5 >> n^3 >> n^2 \,/\, n^2 + \log n >> n \log n >> n$
$>> \frac{n}{\log n} >> \sqrt{n} >> \log n + n^{\frac{1}{3}} >> (\log n)^2 >> \log n \,/\, \ln n >> \log \log n >> 6 >> (\frac{1}{3})^n$

2-20 [5] Find two functions $f(n)$ and $g(n)$ that satisfy the following relationship. If no such f and g exist, write none.

(a) $f(n) = o(g(n))$ and $f(n) = \Theta(g(n))$
(b) $f(n) = \Theta(g(n))$ and $f(n) = o(g(n))$
(c) $f(n) = \Theta(g(n))$ and $f(n) \neq O(g(n))$
(d) $f(n) = \Omega(g(n))$ and $f(n) \neq O(g(n))$

a) By definition:
$f(n) = o(g(n))$ if $f(n) = O\,g(n)$ and $f(n) \neq \Theta(g(n))$ => none

b) none

c) By definition:
If $f(n)$ is $O(g(n))$ then $f(n)$ is $O(g(n))$ and $g(n)$ is $O(f(n))$
or, i.e. $f(n)$ is $O(g(n))$ and $f(n)$ is $\Omega(g(n))$.
=> none

d) $f(n) = \sqrt{n}$ $\quad g(n) = 2 \log n$

$$\lim_{n \to \infty} \frac{\sqrt{n}}{2 \log n} = 2 \lim_{n \to \infty} \frac{\sqrt{n}}{\log n} = \infty$$

$$\therefore \lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty \Rightarrow f(n) = \Omega(g(n))$$

However
$f(n) = O\ g(n)$ iff $\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0.$

Since $\lim_{n \to \infty} \frac{f(n)}{g(n)} \neq 0$ $\quad f(n) \neq O(g(n))$ ✓

Algorithm Design Manual
Skiena - Chapter 2
Exercises

Melissa Auclaire

2-21 [5] True or False?

(a) $2n^2 + 1 = O(n^2) \rightarrow \lim_{n\to\infty} \frac{2n^2+1}{n^2} = 2 \Rightarrow 2n^2 + 1 = \Theta(n^2) \Rightarrow 2n^2 + 1 = O(n^2) \Rightarrow$ True

(b) $\sqrt{n} = O(\log n)$ False (looking at the efficiency class chart) $\sqrt{n} >> \log n$

(c) $\log n = O(\sqrt{n})$ False $\sqrt{n} >> \log n$

(d) $n^2(1+\sqrt{n}) = O(n^2 \log n) \Rightarrow (n^2 + n^{\frac{5}{2}}) = O(n^2 \log n)$ $\lim_{n\to\infty} \frac{n^2 + n^{\frac{5}{2}}}{n^2 \log n} = \infty \Rightarrow$ False

(e) $3n^2 + \sqrt{n} = O(n^2) \rightarrow \lim_{n\to\infty} \frac{3n^2+\sqrt{n}}{n^2} \Rightarrow 3n^2 + \sqrt{n} = \Theta(n^2) \Rightarrow 3n^2 + \sqrt{n} = O(n^2) \Rightarrow$ True

(f) $\sqrt{n} \log n = O(n)$ $n >> \sqrt{n}$ $\therefore$ False

(g) $\log n = O(n^{-\frac{1}{2}}) \rightarrow \lim_{n\to\infty} \frac{\log n}{n^{\frac{1}{2}}} = \infty \Rightarrow \log(n) = \Omega(n^{-\frac{1}{2}})$, $\log(n) \neq \Theta(n^{-\frac{1}{2}})$ $\therefore$ False

2-22 [5] For each of the following pairs of functions $f(n)$ and $g(n)$, state whether $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$, $f(n) = \Theta(g(n))$, or none of the above.

(a) $f(n) = n^2 + 3n + 4$, $g(n) = 6n + 7$
(b) $f(n) = n\sqrt{n}$, $g(n) = n^2 - n$
(c) $f(n) = 2^n - n^2$, $g(n) = n^4 + n^2$

a) $\lim_{n\to\infty} \frac{n^2+3n+4}{6n+7} = \lim_{n\to\infty} \frac{n + 3 + \frac{4}{n}}{6 + \frac{7}{n}} = \lim_{n\to\infty} \frac{n+3}{6} = \lim_{n\to\infty} \frac{n}{6} = \infty$

$\therefore f(n) = \Omega(g(n))$

b) $\lim_{n\to\infty} \frac{n\sqrt{n}}{n^2-n} = \lim_{n\to\infty} \frac{n^{\frac{3}{2}}}{n^2-n}$ | Since: $n^2 >> n^{\frac{3}{2}}$ $\lim_{n\to\infty} \frac{n^{\frac{3}{2}}}{n^2-n} = 0$

$\therefore f(n) = O(g(n))$

c) $\lim_{n\to\infty} \frac{2^n - n^2}{n^4+n^2} = \lim_{n\to\infty} \frac{\frac{2^n}{n^2} - \frac{1}{n^2}}{1 + \frac{1}{n^2}} = \lim_{n\to\infty} \frac{2^n}{n^4}$ | Since $2^n >> n^4$ $\lim_{n\to\infty} \frac{2^n}{n^4} = \infty$

$\therefore f(n) = \Omega(g(n))$

2-23 [3] For each of these questions, briefly explain your answer.

(a) If I prove that an algorithm takes $O(n^2)$ worst-case time, is it possible that it takes $O(n)$ on some inputs?

(b) If I prove that an algorithm takes $O(n^2)$ worst-case time, is it possible that it takes $O(n)$ on all the inputs?

(c) If I prove that an algorithm takes $\Theta(n^2)$ worst-case time, is it possible that it takes $O(n)$ on some inputs?

a) Yes, it is possible. $O(n^2)$ denotes the upper bound (i.e) the worst case which has the longest possible runtime for any input. Some inputs may have considerably lower runtimes, and the average case may be less than the worst-case.

b) Yes, it is possible. The complexity class of n^2 grows faster asymptotically than n, and thus all inputs are within the upper bound of $O(n^2)$, even though this is not the lowest possible upper-bound.

c) Yes, it is possible. Although the algorithm may be $\Theta(n^2)$ in the worst-case, the runtime efficiency can be lower in the average case or for some input(s) n.

Algorithm Design Manual
Skiena - Chapter 2
Exercises

Melissa Auclaire

2-23 (d) If I prove that an algorithm takes $\Theta(n^2)$ worst-case time, is it possible that it takes $O(n)$ on all inputs?

(e) Is the function $f(n) = \Theta(n^2)$, where $f(n) = 100n^2$ for even n and $f(n) = 20n^2 - n\log_2 n$ for odd n?

d) No it isn't. Since $\Theta(n^2)$ implies $\Omega(n^2)$ the runtime efficiency class of all inputs must be >> n.

e) Yes, it is: $f(n) = \Theta(n^2)$ since $f_{even}(n) = \Theta(n^2)$ and $f_{odd}(n) = \Theta(n^2)$.

2-24 [3] For each of the following, answer yes, no, or can't tell. Explain your reasoning.

(a) Is $3^n = O(2^n)$?
(b) Is $\log 3^n = O(\log 2^n)$?
(c) Is $3^n = \Omega(2^n)$?
(d) Is $\log 3^n = \Omega(\log 2^n)$?

a) $\lim_{n\to\infty} \frac{3^n}{2^n} = \infty$

$\lim_{n\to\infty} \left(\frac{3}{2}\right)^n =$

$\left(\frac{3}{2}\right)^{\lim_{n\to\infty} n} =$

$\left(\frac{3}{2}\right)^{\infty} = \infty$

$\therefore f(n)\ \Omega(g(n)) \Rightarrow$ False

$f(n) \le c \cdot g(n)$ for some c when $n \ge n_0$.
Since there is no c & $n > n_0$
which fulfills this condition
⇓

c) $\lim_{n\to\infty} \frac{3^n}{2^n} =$

$\left(\frac{3}{2}\right)^{\lim_{n\to\infty} n} =$

$\left(\frac{3}{2}\right)^{\infty} = \infty \Rightarrow f(n) = \Omega(g(n))$

$\Rightarrow$ True

b) $\lim_{n\to\infty} \frac{\log 3^n}{\log 2^n} =$

$\lim_{n\to\infty} \frac{\frac{d}{dn}(\log(3^n))}{\frac{d}{dn}(\log(2^n))} =$ L'Hopital

$\lim_{n\to\infty} \frac{\log 3}{\log 2}$ $\therefore \log 3^n = \Theta(\log 2^n) \Rightarrow \log 3^n = O(\log 2^n) \Rightarrow$ True

d) $\lim_{n\to\infty} \frac{3^n}{\log 2^n} = \lim_{n\to\infty} \frac{\frac{d}{dn}(3^n)}{\frac{d}{dn}(\log 2^n)} = \lim_{n\to\infty} \frac{3^n \log 3}{\log 2} =$

$\lim_{n\to\infty} \frac{\log(3)3^n}{\log 2} = \frac{\log 3(\lim_{n\to\infty} 3^n)}{\log 2} = \frac{\log 3(3^{\lim_{n\to\infty} n})}{\log 2} =$

Product Rule

$\frac{\log 3(3^{\infty})}{\log 2} = \infty \Rightarrow f(n) = \Omega(g(n))$

$\Rightarrow$ True

2-25 [5] For each of the following expressions $f(n)$ find a simple $g(n)$ such that $f(n) = \Theta(g(n))$.

(a) $f(n) = \sum_{i=1}^{n} \frac{1}{i}$.

(b) $f(n) = \sum_{i=1}^{n} \lceil \frac{1}{i} \rceil$.

(c) $f(n) = \sum_{i=1}^{n} \log(i)$.

(d) $f(n) = \log(n!)$.

a) $\sum_{i=1}^{n} \frac{1}{i} = \frac{\frac{1}{n(n+2)}}{2} = \frac{2}{n(n+1)} = \frac{2}{n^2+n}$ | $\approx \log\left(\frac{2b+1}{2a-1}\right) \Rightarrow \log n$

b) $\sum_{i=1}^{n} \lceil \frac{1}{i} \rceil = \sum_{i=1}^{n} 1 = n \Rightarrow n$

c) $\sum_{i=1}^{n} \log(i) = n\log(i) \Rightarrow n\log n$

d) $\lim_{n\to\infty} \frac{\log(n!)}{n\log n} = 1 \Rightarrow n\log n$

Melissa Auclaire

Algorithm Design Manual
Skiena · Chapter 2
Exercises

2-26 [5] Place the following functions into increasing asymptotic order.
$f_1(n) = n^2 \log_2 n$, $f_2(n) = n(\log_2 n)^2$, $f_3(n) = \sum_{i=0}^{n} 2^i$, $f_4(n) = \log_2\left(\sum_{i=0}^{n} 2^i\right)$.

$f_1(n) = n^2 \log_2 n \mid \lim_{n\to\infty} \frac{n^2 \log_2 n}{n^3} = 0 \mid \lim_{n\to\infty} \frac{\log n}{n \log 2} = \frac{\lim \frac{\log n}{n}}{\log 2} = \frac{1}{\log 2} \lim_{n\to\infty} \frac{\log n}{n} \mid n \gg \log(n) \Rightarrow 0 \Rightarrow$ $f_1(n) = O(n^3)$

$f_2(n) = n(\log_2 n)^2 \mid \lim_{n\to\infty} \frac{n(\log_2 n)^2}{n^2} = 0 \mid \lim_{n\to\infty} \frac{\log^2(n)}{n \log^2(2)} = \frac{\lim_{n\to\infty} \frac{\log^2 n}{n}}{\log^2(2)} = \frac{1}{\log^2 2} \lim_{n\to\infty} \frac{\log^2 n}{n} = n \gg \log^2 n \Rightarrow 0$ $f_2(n) = O(n^2)$

$f_3(n) = \sum_{i=0}^{n} 2^i = 2^{n+1} - 1 \mid \lim_{n\to\infty} \frac{2^{n+1}-1}{2^n} \mid \lim_{n\to\infty} \frac{2^{n+1}-1}{2n} = \lim_{n\to\infty} 2^{-n}(2^{n+1}-1) = \lim_{n\to\infty} \frac{2 - 2^{-n}}{1} = 2 \Rightarrow$ $f_3(n) = \Theta(2^n) \Rightarrow f_3(n) = O(2^n)$

$f_4(n) = \log_2\left(\sum_{i=0}^{n} 2^i\right) = \log_2(2^{n+1} - 1) \mid$

$\lim_{n\to\infty} \frac{\log_2(2^{n+1}-1)}{n} \mid \lim_{n\to\infty} \frac{\log(2^{n+1}-1)}{n \log 2} = \frac{\lim_{n\to\infty} \frac{\log(2^{n+1}-1)}{n}}{\log 2} = \frac{1}{\log 2} \lim_{n\to\infty} \frac{\log(2^{n+1}-1)}{n} =$

$\frac{1}{\log 2} \lim_{n\to\infty} \frac{\frac{d}{dn}\left(\log(2^{n+1}-1)\right)}{\frac{dn}{dn}} = \frac{1}{\log 2} \lim_{n\to\infty} \frac{\frac{2^{n+1}\log 2}{2^{n+1}-1}}{1} = \frac{\lim_{n\to\infty}\left(\frac{2^{n+1}\log 2}{2^{n+1}-1}\right)}{\log 2} = \frac{\log 2 \lim_{n\to\infty}\left(\frac{2^{n+1}}{2^{n+1}-1}\right)}{\log 2} =$

$\frac{\log 2 \lim_{n\to\infty} \frac{1}{1 - 2^{-1-n}}}{\log 2} = \frac{\log 2}{\log 2} = 1 \Rightarrow f_4(n) = \Theta(n) \Rightarrow f_4(n) = O(n)$

$f_1(n) = O(n^3)$ $\quad 2^n \gg n^3 \gg n^2 \gg n$
$f_2(n) = O(n^2)$ $\quad \therefore f_3(n) \gg f_1(n) \gg f_2(n) \gg f_4(n)$
$f_3(n) = O(2^n)$
$f_4(n) = O(n)$

2-27 [5] Place the following functions into increasing asymptotic order. If two or more of the functions are of the same asymptotic order then indicate this.
$f_1(n) = \sum_{i=1}^{n} \sqrt{i}$, $f_2(n) = (\sqrt{n}) \log n$, $f_3(n) = n\sqrt{\log n}$, $f_4(n) = 12^{\frac{3}{2}} + 4n$.

$f_1(n) = \sum_{i=1}^{n} \sqrt{i} = \left(\sum_{i=1}^{n} \frac{1}{i}\right)^{-\frac{1}{2}} \mid$ We know from 2-25 a) that $\sum_{i=1}^{n} \frac{1}{i} = O(\log n)$
$\therefore f_1(n) = O(\log^{-\frac{1}{2}} n) = f_1(n) = O\left(\frac{1}{\sqrt{\log n}}\right)$

$f_2(n) = (\sqrt{n}) \log n \quad \lim_{n\to\infty} \frac{\sqrt{n}\log n}{n} = \lim_{n\to\infty} \frac{\log n}{\sqrt{n}} \quad \sqrt{n} \gg \log n \Rightarrow \lim_{n\to\infty} \frac{\log n}{\sqrt{n}} = 0 \Rightarrow f_2(n) = O(n)$

$f_3(n) = n\sqrt{\log n} \quad \lim_{n\to\infty} \frac{n\sqrt{\log n}}{n} = \lim_{n\to\infty} \sqrt{\log n} = \sqrt{\lim_{n\to\infty} \log n} = \sqrt{\infty} = \infty$
$\lim_{n\to\infty} \frac{n\sqrt{\log n}}{n^2} = \lim_{n\to\infty} \frac{\sqrt{\log n}}{n} \quad n \gg \sqrt{\log n} \Rightarrow \lim_{n\to\infty} \frac{\sqrt{\log n}}{n} = 0 \Rightarrow f_3(n) = O(n^2)$

$f_4(n) = 12^{\frac{3}{2}} + 4n \quad \lim_{n\to\infty} \frac{12^{\frac{3}{2}} + 4n}{n} = \lim_{n\to\infty} \frac{4n+1}{n} = \lim_{n\to\infty} \frac{4 + \frac{1}{n}}{1} = 4 \Rightarrow f_4(n) = \Theta(n) \Rightarrow f_4(n) = O(n)$

$f_1(n) = O\left(\frac{1}{\sqrt{\log n}}\right)$ $\quad n^2 \gg n \gg \frac{1}{\sqrt{\log n}}$
$f_2(n) = O(n)$ $\quad \therefore f_3(n) \gg f_2(n) / f_4(n) \gg f_1(n)$
$f_3(n) = O(n^2)$
$f_4(n) = O(n)$