Melissa Auclaire

Algorithm Design Manual
Skiena - Chapter 2
Notes

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 dominates f $\rightarrow g \gg f$

$\overbrace{g(n) = O(f(n))}^{f \gg g}$ 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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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
```

Algorithm Design Manual
Skiena - Chapter 2
Exercises

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:

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

Melissa Auclaire

Algorithm Design manual
Skiena - Chapter 2
Exercises

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)
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 $c \ge 2$.

∴ true

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

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

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

$\log n + 5 \leq \log n + 5\log n$
$\log n + 5 \leq 6\log n$
$\log n + 5 \leq 3(2)\log(n)$
$3\log n^2 \geq \log n + 5$
$\log n^2 \geq c(\log n + 5)$ 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 $c \mid f(n) \leq c \cdot g(n)$ 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) $1^2 + 1 + 1 \overset{?}{\leq} 2(1)^3$

$3 \not\leq 2$

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

$7 \leq 16$

$\therefore c = 1$

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

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

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

$1 \leq 2$

$\therefore c = 1$

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

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

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

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

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

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

$\therefore c = 3$

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 $f_1(n) + f_2(n) = \Omega(g_1(n) + g_2(n))$.

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

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 $k = 2$
$a_{k1} n^{k+1} + a_k n^k + \dots + a_1 n + a_0 = O(n^{k+1})$
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

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

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

$34 \leq 40$ $\quad 40 - 34 = 6$

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

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

$81 + 18 + 3 \leq 135$

$102 \leq 135$ $\quad 135 - 102 = 33$

$\therefore a_{k+1}n^{k+1} + a_k n^k + \ldots + a_1 n + a_0 \leq c \cdot n^{k+1}$ 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: $(a = 0, b = 1)$

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

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

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

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

e.x. $f(n) \leq c_1 \cdot g(n)$ $\quad c_1 = 1, c_2 = 1, n \geq 0$

$g(n) \geq c_2 \cdot g(n)$ $\quad f(n) = g(n)$

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.

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

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

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

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

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

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:
$f(n) = o(g(n))$ iff $f(n) = O\, g(n)$ and $f(n) \neq \Theta(g(n))$ => none

b) none

c) By definition:
If $f(n)$ is $\Theta(g(n))$ 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) $\lim_{n\to\infty} \frac{2^n - n^2}{n^4+n^2} = \lim_{n\to\infty} \frac{\frac{2^n}{n^4} - \frac{1}{n^2}}{1 + \frac{1}{n^2}} = \lim_{n\to\infty} \frac{2^n}{n^4}$ | 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.

Melissa Auclaire

Algorithm Design Manual
Skiena - Chapter 2
Exercises

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
$\Downarrow$

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

$\lim_{n\to\infty} \frac{\log(3)3^n}{\log 2} = \frac{\log 3(\lim_{n\to\infty} 3^n)}{\log 2} = \frac{\log 3(\lim_{3n\to\infty} n)}{\log 2} =$ 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) $\sum_{i=1}^{n} \frac{1}{i} = \frac{\frac{1}{n(n+2)}}{2} = \frac{2}{n(n+2)} = \frac{2}{n^2+n}$ $\Big|$ $\approx \log\left(\frac{2b+1}{2a-1}\right) \Rightarrow \log n$

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.
$f_1(n) = n^2\log_2 n$, $f_2(n) = n(\log_2 n)^2$, $f_3(n) = \sum_{i=0}^{n} 2^i$, $f_4(n) = \log_2(\sum_{i=0}^{n} 2^i)$.

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

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

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

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

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

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

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

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

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 $\sum_{i=1}^{n}\frac{1}{i} = O(\log n)$
$\therefore f_1(n) = O(\log^{-\frac{1}{2}} n) = f_1(n) = O\left(\frac{1}{\sqrt{\log n}}\right)$

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

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

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

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

Algorithm Design Manual
Skiena - Chapter 2
Exercises

Melissa Auclaire

2-28 [5] For each of the following expressions $f(n)$ find a simple $g(n)$ such that $f(n) = \Theta(g(n))$. (You should be able to prove your result by exhibiting the relevant parameters, but this is not required for the homework).

(a) $f(n) = \sum_{i=1}^{n} 3i^4 + 2i^3 - 19i + 20$. $\approx \sum_{i=1}^{n} 3i^4$ as $n \to \infty$

(b) $f(n) = \sum_{i=1}^{n} 3(4^i) + 2(3^i) - i^{19} + 20 \approx \sum_{i=1}^{n} 3(4^i) + 2(3^i)$ as $n \to \infty$

(c) $f(n) = \sum_{i=1}^{n} 5^i + 3^{2i} \approx \sum_{i=1}^{n} 3^{2i}$ as $n \to \infty$

a) $f(n) = \Theta(n^5)$
b) $f(n) = \Theta(2^{2n})$ (Proofs attached as corollary 1)
c) $f(n) = \Theta(9^n)$

2-29 [5] Which of the following are true?

(a) $\sum_{i=1}^{n} 3^i = \Theta(3^{n-1})$.

(b) $\sum_{i=1}^{n} 3^i = \Theta(3^n)$.

(c) $\sum_{i=1}^{n} 3^i = \Theta(3^{n+1})$.

a) $\sum_{i=1}^{n} 3^i = \frac{3}{2}(3^n - 1)$ $\lim_{n\to\infty} \frac{\frac{3}{2}(3^n-1)}{3^{n-1}} = \lim_{n\to\infty} \frac{1}{2} 3^{2-n}(3^n - 1) =$

$\frac{\lim_{n\to\infty} 3^{2-n} 3^n}{2} = \frac{1}{2}(9) = 9 \Rightarrow \sum_{i=1}^{n} 3^i = \Theta(3^{n-1}) \therefore$ True

b) $\sum_{i=1}^{n} 3^i = \frac{3}{2}(3^n - 1)$ $\lim_{n\to\infty} \frac{\frac{3}{2}(3^n-1)}{3^n} = \lim_{n\to\infty} \frac{3(3^n-1)}{(2)(3^n)} = \lim_{n\to\infty} \frac{1}{2} 3^{1-n}(3^n - 1) =$

$\frac{1}{2}\left(\lim_{n\to\infty} 3^{1-n}(3^n - 1)\right) = \lim_{n\to\infty} \frac{3^{1-n} 3^n}{2} = \left(\frac{1}{2}\right)3 = \frac{3}{2} \Rightarrow \sum_{i=1}^{n} 3^i = \Theta(3^n) \therefore$ True

c) $\sum_{i=1}^{n} 3^i = \frac{3}{2}(3^n - 1)$ $\lim_{n\to\infty} \frac{\frac{3}{2}(3^n-1)}{3^{n+1}} = \lim_{n\to\infty} \frac{3^n - 1}{(2)(3^n)} = \lim_{n\to\infty} \frac{1}{2} 3^{-n}(3^n - 1) =$

$\frac{1}{2} \lim_{n\to\infty} 3^{-n}(3^n - 1) = \lim_{n\to\infty} \frac{\frac{1 - 3^{-n}}{1}}{2} = \frac{1}{2} \Rightarrow \sum_{i=1}^{n} 3^i = \Theta(3^{n+1}) \therefore$ True

Big Θ Def.
$\forall n \geq n_0 \ \exists (C_1, C_2)$
$| f(n) \leq C_1 \cdot g(n)$ and
$f(n) \geq C_2 \cdot g(n)$

2-30 [5] For each of the following functions f find a simple function g | $f(n) = \Theta(g(n))$.

(a) $f_1(n) = (1000)2^n + 4^n$
(b) $f_2(n) = n + n\log n + \sqrt{n}$
(c) $f_3(n) = \log(n^{20}) + (\log n)^{10}$
(d) $f_4(n) = (0.99)^n + n^{100}$

a) $2000^n + 4000^n$ $n_0 = 1$

$2000^n + 4000^n \leq 6000^n \cdot 2$
$2000^n + 4000^n \geq 6000^n \cdot 1$
so theorem holds for $n_0 \geq 0$, $C_1 = 2$, $C_2 = 1$
$\therefore f_1(n) = \Theta(6000^n)$

$n\log n > n^{\frac{3}{2}} = n\sqrt{n}$

b) $n + n\log n + \sqrt{n} \leq n\log n \cdot 2$
$n + n\log n + \sqrt{n} \geq n\log n \cdot 1$
so theorem holds for $n_0 \geq 0$
$\therefore f_2(n) = \Theta(n\log n)$

Algorithm Design Manual
Skiena - Chapter 2
Exercises

Melissa Auclaire

Big Θ Def
$\forall n \geq n_0 \ \exists (c_1, c_2) \mid$
$f(n) \leq c_1 \cdot g(n)$ and
$f(n) \geq c_2 \cdot g(n)$

2-30 [5] contd.

c) $\log(n^{20}) + (\log n)^{10} \leq \log^{10}(n^{20}) \cdot 2$
$\log(n^{20}) + (\log n)^{10} \geq \log^{10}(n^{20}) \cdot 1$
So the theorem holds for $n_0 \geq 1$
$\therefore f_3(n) = \Theta(\log^{10}(n^{20}))$

d) $(0.99)^n + n^{100} \leq n^{100} \cdot 2$ since $0.99^n < n^{100} \ \forall n_0 \geq 1$
$(0.99)^n + n^{100} \geq n^{100} \cdot 1$
So the theorem holds for $n_0 \geq 1$
$\therefore f_4(n) = \Theta(n^{100})$

2-31 [5] For each pair of expressions (A, B) below, indicate whether A is O, o, Ω, ω, or Θ of B. Note that zero, one or more of these relations may hold for a given pair; list all the correct ones.

	A	B
(a)	n^{100}	2^n
(b)	$(\log n)^{12}$	$\sqrt{n}$
(c)	$\sqrt{n}$	$n^{\cos(\pi n/8)}$
(d)	10^n	100^n
(e)	$n^{\lg n}$	$(\log n)^n$
(f)	$\log(n!)$	$n \log n$

Little Oh Def
$f(n)$ is $o(g(n))$ if $f(n) = O(g(n))$ and $f(n) \neq \Theta(g(n))$

Little Omega Def
$f(n)$ is $\omega(g(n))$ if $f(n) = \Omega(g(n))$ and $f(n) \neq \Theta(g(n))$

a) $c^n >> n^{100}$
$$\lim_{n\to\infty} \frac{n^{100}}{2^n} = \lim_{n\to\infty} 2^{-n} n^{100} = 0 \Rightarrow A = O(B), \ A = o(B)$$

b) $\sqrt{n} >> \log^2 n$
$$\lim_{n\to\infty} \frac{\log^{12} n}{\sqrt{n}} = 0 \Rightarrow A = O(B), \ A = o(B)$$

c) $\sqrt{n} >> n^{1+\epsilon}$
$$\lim_{n\to\infty} \frac{\sqrt{n}}{n^{\cos(\pi n/8)}} \text{ DNE However } \sqrt{n} >> n^{1+\epsilon} \Rightarrow A = \Omega(B), \ A = \omega(B)$$

d) $c^n = c^n$
$$\lim_{n\to\infty} \frac{10^n}{100^n} = \lim_{n\to\infty} 10^{-n} = \frac{1}{\lim_{n\to\infty} 10^n} = \frac{1}{10^{\lim_{n\to\infty} n}} = \frac{1}{10^\infty} = 0 \Rightarrow A = O(B), \ A = o(B)$$

e) $2^n >> n^{\log(n)}$ $n >> \log(n)$
$$\lim_{n\to\infty} \frac{n^{\log n}}{\log^n n} = \lim_{n\to\infty} n^{\log n} \log^{-n} n = 0 \Rightarrow A = O(B), \ A = o(B)$$

f) $\lim_{n\to\infty} \frac{\log(n!)}{n \log n}$ $\ln(n!) = \sum_{k=1}^{n} \ln k \sim \int_1^n \ln dx = n\ln n - n + 1$ $\lim_{n\to\infty} \frac{n \ln n - n + 1}{n \log n} =$

Algorithm Design Manual
Skiena - Chapter 2
Exercises

Melissa Auclaire

2-31 [5] contd

$$\lim_{n\to\infty} \frac{n\ln n - n + 1}{n\log n} = \lim_{n\to\infty} \frac{n\log n + n - 1}{n\log n} = \lim_{n\to\infty} \frac{\frac{1}{n\log n} - \frac{1}{\log n} + 1}{1} = 1 \Rightarrow A = \Theta(B),\ A = O(B),\ A = \Omega(B)$$

Summations

2-32 [5] Prove that:

$$1^2 - 2^2 + 3^2 - 4^2 + \dots + (-1)^{k-1}k^2 = (-1)^{k-1}k(k+1)/2$$

$$1^2 - 2^2 + 3^2 - 4^2 + \dots + (-1)^{k-1}k^2 = -\tfrac{1}{2}(-1)^k k(k+1)$$

$$-\tfrac{1}{2}(-1)^k k(k+1) = \sum_{n=1}^{k}(-1)^{n-1}n^2$$

Base case:

$k = 0$

$-\frac{1}{2}(-1)^0\, 0(0+1)$

$1 \cdot 0(1) = 0$ ✓

$k = 1$

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

$\frac{1}{2} \cdot 2 = 1 = 1^2$ ✓

Inductive step: $k = k+1$

$$-\tfrac{1}{2}(-1)^{k+1} \cdot (k+1)((k+1)+1) = 1^2 - 2^2 + 3^2 - 4^2 + \dots + (-1)^k(k+1)^2$$

$$\frac{(-1)^k \cdot (k+1)(k+2)}{2} = 1^2 - 2^2 + 3^2 - 4^2 + \dots + (-1)^k(k+1)^2$$

$$\tfrac{1}{2}(-1)^k(k+1)(k+2) = 1^2 - 2^2 + 3^2 - 4^2 + \dots + (-1)^k(k^2+2k+1)$$

$$\tfrac{1}{2}(-1)^k(k+1)(k+2) = \sum_{k=0}^{n}(-1)^n(n+1)^2$$

2-33 [5] Find an expression for the sum of the ith row of the following triangle, and prove its correctness. Each entry is the sum of the three entries directly above it. All non-existing entries are considered 0.

1	= 1	Expression
1 1 1	= 3	3^{i-1}
1 2 3 2 1	= 9	
1 3 6 7 6 3 1	= 27	$3^0 = 1,\ 3^1 = 3,\ 3^2 = 9,\ 3^4 = 27,\ 3^5 = 81$
1 4 10 16 19 16 10 4 1	= 81	

2-34 [3] Assume that Christmas has n days. Exactly how many presents did my "true love" send me? (Do some research if you do not understand this question).

$n = 12$

$12 + 11 + 10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1$

In General

$\sum_{k=1}^{n} k$ = # of presents

2-35 [5] Consider the following code fragment.

```
for i = 1 to n do
    for j = i to 2*i do
        output "foobar"
```

a) $\sum_{i=1}^{n}\sum_{j=i}^{2*i} 1$

b) $T(N) = \sum_{i=1}^{n}\sum_{j=i}^{2*i} 1 = \sum_{i=1}^{n}(2i - i + 1) = \sum_{i=1}^{n}(i+1) = \frac{n(n+1)}{2} + n$

Let $T(N)$ denote the number of times 'foobar' is printed as a function of n.

a. Express $T(n)$ as a summation (actually two nested summations).

b. Simplify the summation. Show your work.

Algorithm Design Manual
Skiena - Chapter 2
Exercises

Melissa Auclaire

2-36 [5] Consider the following code fragment.

```
for i=1 to n/2 do
    for j=i to n-i do
        for k=1 to j do
            output "foobar"
```

Assume n is even. Let T(n) denote the number of times 'foobar' is printed as a function of n.

(a) Express T(n) as three nested summations.

(b) Simplify the summation. Show your work.

a) b)

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

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

$$\sum_{i=1}^{\frac{n}{2}} -in + \sum_{i=1}^{\frac{n}{2}} \frac{n^2}{2} + \sum_{i=1}^{\frac{n}{2}} \frac{n}{2} = -\frac{1}{8}n^2(n+2) + \frac{n^3}{4} + \frac{n^2}{4} = -\frac{1}{8}(n^3+2n^2) + \frac{n^3}{4} + \frac{n^2}{4} =$$

$$-\frac{n^3}{8} - \frac{2n^2}{8} + \frac{n^3}{4}\left(\frac{2}{2}\right) + \frac{n^2}{4}\left(\frac{2}{2}\right) = -\frac{n^3}{8} - \cancel{\frac{2n^2}{8}} + \frac{2n^3}{8} + \cancel{\frac{2n^2}{8}} = \frac{n^3}{8}$$

2-37 [6] When you first learned to multiply numbers, you were told that x × y means adding x a total of y times, so 5 × 4 = 5 + 5 + 5 + 5 = 20. What is the time complexity of multiplying two n-digit numbers in base b (people work in base 10, of course, while computers work in base 2) using the repeated addition method, as a function of n and b. Assume that single-digit by single-digit addition or multiplication takes O(1) time. (Hint: How big can y be as a function of n and b?)

multiplu(a, c) n = digits b = base $\overset{a}{10} \times \overset{c}{15}$ O(n)

multiply(a,c) = $O(a^n)$

(n-1) additional mults

(an-n) adds per mult

$n(an-n) = an^2-n^2$

Place a's for each place or n-i a's where i is the current place

=> multiply(a,c) = $O(an^2-n^2)$

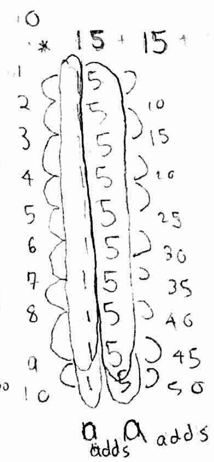
☆ Ask about it!

☆ Ask about it!

(n-1) additional mults

(a-1) n adds per multiply

← and so on...



Single digit add O(1)
Single digit mult O(1)

1's	10's	100's
10 adds	10 adds	10 adds
0 mults	10 mults	10 mults

If two digits or more:

10 → 10/2, 10/2

n = 2 b = 16

y ≤ 99 × 99 = 9801

y ≤ a × c

Algorithm Design Manual
Skiena - Chapter 2
Exercises

Melissa Auclaire

2-38 [6] In grade school, you learned to multiply long numbers on a digit-by-digit basis, so that $127 \times 211 = 127 \times 1 + 127 \times 10 + 127 \times 200 = 26,397$. Analyze the time complexity of multiplying two n-digit numbers with this method as a function of n (assume constant base size). Assume that single-digit by single-digit addition or multiplication takes $O(1)$ time.

multiply(a, c)

multiply(a, c) $= O(a(n-1)10)$

n digits

$127 \times 1 = 1 + 1 + \ldots + 1$
a times

127×10
a 10 times

127×100

a(100) times

Calling recursively

multiply(127, 1)

127×1

127×10

$127 \times 0 + 127 \times 10$

Logarithms

2-39 [5] Prove the following identities on logarithms:

(a) Prove that $\log_a xy = \log_a x + \log_a y$
(b) Prove that $\log_a x^y = y \log_a x$
(c) Prove that $\log_a x = \frac{\log_b x}{\log_b a}$
(d) Prove that $x^{\log_b y} = y^{\log_b x}$

a) $\log_a xy = (\log_a(x) + \log_a(y))$

$xy = a^{\log_a(x) + \log_a y}$

$(a^{\log_a(xy)} \ldots \log_a x + \log_a y)$

$a^m = x \Rightarrow \log_a(x) = m$

$a^n = y \Rightarrow \log_a(y) = n$

$a^l = xy \Rightarrow \log_a(xy) = l$

$a^l = a^m + a^n = a^{m+n}$

$a^l = a^{m+n} \Rightarrow \log_a(xy) = \log_a(x) + \log_a(y)$ ∎

$l = m + n$

b) $\log_a(x^y) = y \log_a(x)$

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

$\log_a(a^{y \log_a(x)}) = y \log_a x$ ∎

c) $\log_a(x) = \frac{\log_b x}{\log_b a}$

$x = a^{\frac{\log_b x}{\log_b a}}$

$\log_a(a^{\frac{\log_b x}{\log_b a}}) = \frac{\log_b x}{\log_b a}$ ∎

d) $x^{\log_b y} = y^{\log_b x}$

$\log_b(x^{\log_b y}) = \log_b(y^{\log_b x})$

$b^{\log_b(y^{\log_b x})} = b^{\log_b(x^{\log_b y})}$

$\log_b(b^{\log_b(y^{\log_b x})}) = \log_b(b^{\log_b(x^{\log_b y})})$

$\log_b(y^{\log_b x}) = \log_b(x^{\log_b y})$

$b^{y^{\log_b x}} = b^{x^{\log_b y}}$

$\log_b(b^{y^{\log_b x}}) = \log_b(b^{x^{\log_b y}})$

$y^{\log_b x} = x^{\log_b y}$ ∎

Algorithm Design Manual
Skiena - Chapter 2
Exercises

Melissa Auclapr

2-40 [3] Show that $\lceil \lg(n+1) \rceil = \lfloor \lg n \rfloor + 1$

Base case : $n = 0$

$\lceil \lg(0+1) \rceil = \lceil \lg(1) \rceil = \lceil 0 \rceil = 0$ ★ Ask about this

$\lfloor \lg 1 \rfloor + 1 = 0 + 1 = 1$

$n = 1$ case

$\lceil \lg(1+1) \rceil = \lceil \lg(2) \rceil = \lceil 0.30 \rceil = 1$

$\lfloor \lg 1 \rfloor + 1 = 0 + 1 = 1$

for $n \geq 1$, $x = 2^k$ for some $k \geq 1$

$2^k < x + 1 < 2^{k+1}$

$k < \log(x+1) < k+1$ while $\lg x = k$

for $n \geq 2$, $2^k < x < 2^{k+1} - 1$ for some $k \geq 1$

$2^k < \log(x) < 2^{k+1} - 1$ while $\lg x = k$

for $n \geq 2$, $x = 2^k - 1$ for some $k \geq 2$

$x < 2^k < 2^{k+1}$ while $\lg x = k$

2-41 [3] Prove that the binary representation of $n \geq 1$ has $\lfloor \lg_2 n \rfloor + 1$ bits.

Base case

$\lg 1 = 0$	$\lfloor \lg 1 \rfloor = 0 + 1 = 1$ bits
$\lg 2 \approx 0.631$	$\lfloor \lg 2 \rfloor = 0 + 1 = 1$ bits
$\lg 3 \approx 1.0986$	$\lfloor \lg 3 \rfloor = 1 + 1 = 2$ bits
$\lg 4 \approx 1.386$	$\lfloor \lg 4 \rfloor = 1 + 1 = 2$ bits
$\lg 5 \approx 1.609$	$\lfloor \lg 5 \rfloor = 1 + 1 = 2$ bits
$\lg 6 \approx 1.792$	$\lfloor \lg 6 \rfloor = 1 + 1 = 2$ bits
$\lg 7 \approx 1.946$	$\lfloor \lg 7 \rfloor = 1 + 1 = 2$ bits
$\lg 8 \approx 2.079$	$\lfloor \lg 8 \rfloor = 2 + 1 = 3$ bits
$\lg 9 \approx 2.197$	$\lfloor \lg 9 \rfloor = 2 + 1 = 3$ bits
$\lg 10 \approx 2.303$	$\lfloor \lg 10 \rfloor = 2 + 1 = 3$ bits

$\lfloor \lg_2(n+1) + 1 \rfloor = \lfloor \lg_2(n+2) \rfloor$

$2^{\lfloor \lg_2(n+1) + 1 \rfloor} = \lfloor \lg_2(n+2) \rfloor$

$b = 2^{n-1} + a_{n-2} 2^{n-2} + \ldots + a_1 2^1 + a_0 2^0$, $a \in [0, 1]$

$2^{n-1} < b < 2^n \Rightarrow n - 1 < \log_2 b < n$

$\lfloor \log_2 b \rfloor = n - 1$ $\therefore \lfloor \log_2 b \rfloor + 1 = b$ ∎

2-42 [5] In one of my research papers I give a comparison-based sorting algorithm that runs in $O(n \log(\sqrt{n}))$. Given the existence of an $\Omega(n \log n)$ lower bound for sorting, how can this be possible?

$n \log \sqrt{n}$ and $n \log n$ are big theta of each other and are in the same efficiency class.

i.e. Changing n to $\sqrt{n}$ does not cause $n \log n$ to dominate $n \log(\sqrt{n})$.

Algorithm Design Manual
Skiena - Chapter 2
Exercises

Melissa Auclaire

Interview Questions

2-43 [5] You are given a set S of n numbers. You must pick a subset S' of k numbers from S such that the probability of each element of S occuring in S' is equal (i.e. each is selected with probability k/n). You may make only one pass over the numbers. What if n is unknown?

$S = \{a_0, a_1, a_2, \ldots, a_n\}$ $k \leq n$ array of n   arr = arr $\{e_0, e_1, e_2, \ldots, e_n\}$

$S' = \{a_0, a_1, a_2, \ldots, a_k\}$

for i := 0 to n

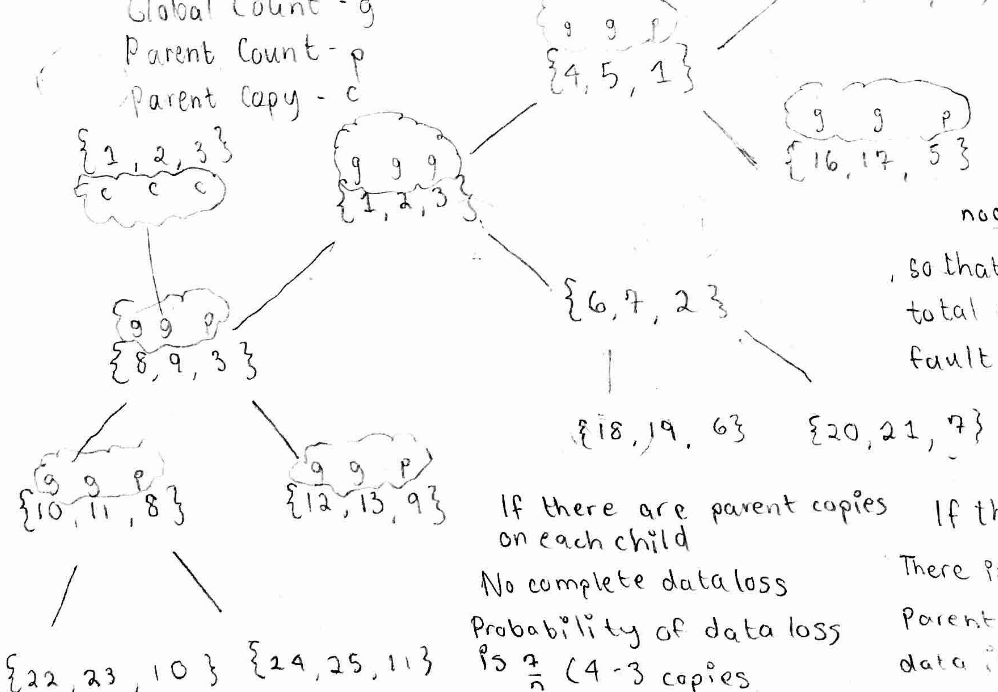
When looping through S, we count the instances of each number in S by incrementing the value of the index of the given number by 1 in the array. Once arr has these numbers, we can create a half subset S' by dividing each number in arr by 2, and adding a number at each index into S' and decrementing the value at that index by 1.

If we don't know n, we can use a dynamically sizing array, or a data structure with dynamic resizing.

2-44 [5] We have 1000 data items to store on 1000 nodes. Each node can store copies of exactly three different items. Propose a replication scheme to minimize data loss as nodes fail. What is the expected number of data entries that get lost when three random nodes fail?

Data Replication Scheme

Global Count - g
Parent Count - p
Parent Copy - c



Start a global count. Have each child of a node contain 2 new data items and one copy of the parent's information. A random child node maintains a copy of the parent, so that each data item is replicated a total of three times. To be more fault tolerant, each child may contain a copy of the parent.

If there are parent copies on each child

No complete data loss

Probability of data loss is $\frac{7}{n}$ (4 - 3 copies, 1 individual data item stored)

If there is 1 parent copy

There is 3 data items, 2 copies of parent node and 1 copy of each data item in other nodes.

Probability of data loss is $\frac{5}{n}$

Prob of complete data loss of 1 node is $\frac{1}{n}$

Algorithm Design Manual
Skiena - Chapter 2
Exercises

Melissa Auclaire

2-45 [5] Consider the following algorithm to find the minimum element in an array of numbers A[0, ..., n]. One extra variable tmp is allocated to hold the current minimum value. Start from A[0]: "tmp" is compared against A[1], A[2], ..., A[N] in order. When A[i] < tmp, tmp = A[i]. What is the expected number of times that the assignment operation tmp = A[i] is performed?

$\sum_{i=1}^{n} \frac{1}{i}$ Average number of assignments for i elements in an unsorted array

2-46 [5] You have a 100 story building and a couple of marbles. You must identify the lowest floor for which a marble will break if you drop it from this floor. How fast can you find this floor if you are given an infinite supply of marbles? What if you have only two marbles?

Infinite supply: lg n tries, drop from $\frac{n}{2}$. If it breaks, drop at the halfway point between 0 and $\frac{n}{2}$. Otherwise, drop at the halfway between $\frac{n}{2}$ and 1. Repeat this process successively until the floor is found.

2 marbles: $\frac{n}{2}$ tries at most. Drop the first marble at $\frac{n}{2}$. If it breaks, begin dropping the second marble from floor 0 to floor $\frac{n}{2} - 1$ until it breaks. The highest floor is floor $m-1$ where m is the floor the marble is dropped and broken. Conversely, if the first marble does not break, count from $\frac{n}{2} + 1$ to n instead, and repeat the above process for $\frac{n}{2} + 1 \leq m \leq n$.

2-47 [5] You are given 10 bags of gold coins. Nine bags contain coins that each weigh 10 grams. One bag contains all false coins that weigh one gram less. You must identify this bag in just one weighing. You have a digital balance which reports the weight of what is placed on it.

Take 1 coin out of each bag and label it with a number that corresponds to the bag from which it was removed. Place all coins on the digital balance. Remove 1 coin at a time in succession, taking note of the balance weight. When the weight of the coin removed is 9 grams, this is the bag which contains the false coins.

2-48 [5] You have eight balls all of the same size. Seven of them weigh the same, and one of them weighs slightly more. How can you find the ball that is heavier by using a balance and only two weighings?

Weigh 3 balls on each side. If the scale balances, weigh the remaining 2 balls. The heavier ball will be on either side. If the scale does not balance, remove the 3 balls on the heavier side. Weigh two of these balls. If the scale balances, the heavier ball was the ball not weighed. If the scale is unbalanced, the heavier ball will be on either side.

Algorithm Design Manual Melissa Auclaire

Skiena, Ch 2

Exercises

2-49 [5] Suppose we start with n companies that eventually merge into one big Company. How many different ways are there for them to merge?

$a_0 + a_1 + \dots + n-1$: = Total different ways for companies to merge

(1,3) (1,2) (1,4) ~~(2,1)~~ (2,3) (2,4) ~~(3,1)~~ ~~(3,2)~~ (3,4) ~~(4,1)~~ ~~(4,2)~~ ~~(4,3)~~

1 2 3 4

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

n = number of companies

2-50 [5] A Ramanujam Number can be written 2 different ways as the sum of 2 cubes - i.e., there exist distinct a, b, c and d such that $a^3 + b^3 = c^3 + d^3$. Generate all Ramanujam numbers where a, b, c, d < n.

$1^3 = 1$
$2^3 = 8$
$3^3 = 27$
$4^3 = 64$
$5^3 = 125$
$6^3 = 216$
$7^3 = 343$
$8^3 = 512$
$9^3 = 729$
$10^3 = 1000$

Brute force:

```
for a:= 1 to n
    for b:= 1 to n
        for c:= 1 to n
            for d:= 1 to n
                if (a³ - b³) = (c³ - d³) and a, b, c, d are distinct
                print ("Ramanujam Number: " + (a³+b³))
```

2-51 [7] Six pirates must divide $300 among themselves. The division is to proceed as follows. The senior pirate proposes a way to divide the money. Then the pirates vote. If the senior pirate gets at least half the votes he wins, and that division remains. If he doesn't, he is killed and then the next senior-most pirate gets a chance to do the division. Now you have to tell what will happen and why (i.e., how many pirates survive and how the division is done)? All the pirates are intelligent and the first priority is to stay alive and the next priority is to get as much money as possible.

The first pirate can bribe the 5th, 3rd and 1st pirates. If he does not, then the even pirates can can bribe the odds until there are 2 left. In that case (the senior-most) will take all of the cash, and the other pirate will live with no money due to a tie vote. If the 6th pirate bribes pirates 5, 3 and 1, he can give 100 to each.

2-52 [7] Reconsider the pirate problem above, where only one indivisible dollar is to be divided. Who gets the dollar and how many are killed.

The 6th pirate can give the dollar to the first pirate. Since the 5th pirate will be killed if the 6th pirate gets killed, he will vote for the 6th pirate, which gives half the vote.

Melissa Auclaire
Corollary 1
2-28 Proofs

2-28 (a) $n(n+1)(2n+1)(3n^2+3n-1)\frac{1}{10} = \sum_{i=1}^{n} 3i^4 \ldots$

$(n^2+n)(2n+1)(3n^2+3n-1)\frac{1}{10}$

$(2n^3+n^2+2n^2+n)(3n^2+3n-1)\frac{1}{10}$

$(2n^3+3n^2+n)(3n^2+3n-1)\frac{1}{10}$

$6n^5+6n^4-2n^3+9n^4+9n^3-\cancel{3n^2}+3n^3+\cancel{3n^2}-n$

$(6n^5+9n^4+6n^4-2n^3+9n^3+3n^3-3n^2+3n^2-n)\frac{1}{10}$

$(6n^5+15n^4+10n^3-n)\frac{1}{10}$

$\frac{6n^5}{10}+\frac{15n^4}{10}+\frac{10n^3}{10}-\frac{n}{10}$

$$\lim_{n\to\infty}\left(\frac{\frac{6n^5+15n^4+10n^3-n}{10}}{n^5}\right)$$

$$\lim_{n\to\infty}\left(\frac{\frac{6+\frac{15}{n}+\frac{10}{n^2}-\frac{1}{n^4}}{10}}{1}\right) = \frac{3}{5} \Rightarrow \sum_{i=1}^{n} 3i^4 = \Theta(n^5)$$

2-28 (b) $3^{n+1}+4^{n+1}-7 = \sum_{i=1}^{n} 3(4^i)+2(3^i)$

$$\lim_{n\to\infty}\frac{3^{n+1}+4^{n+1}-7}{n!} = 0 \Rightarrow \sum_{i=1}^{n} 3(4^i)+2(3^i) = O(n!)$$

$$\lim_{n\to\infty}\frac{3^{n+1}+4^{n+1}-7}{2^{n+n}} = \lim_{n\to\infty}\frac{3^{n+1}+4^{n+1}-7}{2^{2n}} = \lim_{n\to\infty} 2^{-2n}(3^{n+1}+4^{n+1}-7) =$$

$$\lim_{n\to\infty}\frac{2^{-2n}3^{n+1}-7(2^{-2n})+4}{1} = 4 \Rightarrow \sum_{i=1}^{n} 3(4^i)+2(3^i) = \Theta(2^{n+n}) = \Theta(2^{2n})$$

2-28 (c) $\frac{9}{8}(9^n-1) = \sum_{i=1}^{n} 3^{2i}$

$$\lim_{n\to\infty}\frac{\frac{9}{8}(9^n-1)}{9^n} = \lim_{n\to\infty}\frac{1}{8}9^{1-n}(9^n-1) = \frac{1}{8}\left(\lim_{n\to\infty}9^{1-n}(9^n-1)\right) = \frac{1}{8}\lim_{n\to\infty}9^{1-n}(9^n-1) =$$

$$\frac{\lim_{n\to\infty}9^{1-n}9^n}{8} = \frac{1}{8}(9) = \frac{9}{8} \Rightarrow \sum_{i=1}^{n} 3^{2i} = \Theta(9^n)$$