CMPS 211
Assignment 3 - Solution

Section 3.2

2 - a) $f(x) = 17x + 11$; yes $C = 1$, $k = 11$

c) $f(x) = x\log x$; yes $C = 1$ $k = 1$

d) $f(x) = x^4{=}2$; no

e) $f(x) = 2^x$; no

8- b) $f(x) = 3x^5 + (\log x)^4$

$3x^5 \leq 3x^5$ for $x > 1$

$\log x < x$ for $x > 1$

$\Rightarrow 3x^5 + (\log x)^4 < 3x^5 + (x)^4 = 3x^5 + x^4 < 3x^5 + x^5 = 4x^5$

$\Rightarrow f(x) = O(x^5)$ for $c = 4$ and $x > 1$

c) $f(x) = \dfrac{x^4 + x^2 + 1}{x^4 + 1}$

$x^4 + x^2 + 1 \leq x^4 + x^4 + x^4 = 3x^4$ for $x > 1$

$x^4 + 1 \geq x^{-4}$ for $x > 1$ $\Rightarrow$ $\dfrac{1}{x^4 + 1} \leq x^4$ for $x > 1$

$\Rightarrow \dfrac{x^4 + x^2 + 1}{x^4 + 1} \leq \dfrac{3x^4}{x^4} = 3$

$\Rightarrow f(x) = O(x^0) = O(1)$ for $c = 3$ and $x > 1$

14- a) Is $x^3 = O(x^2)$? **NO**

Want to show that there doesn't exist a pair of c and k to satisfy the equation

$x^3 \leq c\,x^2$ and $x > k$

Since x is positive, we can divide both sides of the inequality by x, then

$x^2 \leq c\,x$ and $x > k$

but there are no values of c and k to satisfy this inequality, if we set $k = 1$.

$\Rightarrow x^3 \neq O(x^2)$.

d) Is $x^3 = O(x^2 + x^4)$? **YES**

$x^3 \leq x^4$ for $x > 1$

$\Rightarrow x^3 \leq x^4 + x^2$ for $x > 1$ since $x^2 > 0$

$\Rightarrow x^3 \leq x^4 + x^4 = 2x^4$ for $x > 1$ since $x^2 \leq x^4$

$\Rightarrow x^3 = O(x^2 + x^4)$ for $c = 2$ and $x > 1$

f) Is $x^3 = O(x^3/2)$? **YES**

$x^3 \le x^3 = 2(x^3/2)$ for $x > 1$

$\Rightarrow x^3 \le 2(x^3/2)$ for $x > 1$

$\Rightarrow x^3 = O(x^3/2)$ for $c = 2$ and $x > 1$

18- Show that $1^k + 2^k + \ldots + n^k = O(n^{k+1})$

$1^k \le n^k$ for $n \ge 1$

$2^k \le n^k$ for $n \ge 1$

$\vdots$

$n^k \le n^k$ for $n \ge 1$

$\Rightarrow 1^k + 2^k + \ldots + n^k \le n^k + n^k + \ldots + n^k$ (n times)

$\Rightarrow 1^k + 2^k + \ldots + n^k \le n(n^k) = n^{k+1}$

$\Rightarrow 1^k + 2^k + \ldots + n^k = O(n^{k+1})$ for $c = 1$, $n \ge 1$ and $k > 0$

24- a) Show $3x + 7 = \Theta(x)$.

1. $3x + 7 \le 4x$ since $7 \le x$ for $x > 7$ $\Rightarrow$ $3x + 7 = O(x)$ for $x > 7$
2. $x \le 3x + 7$ since x is positive $\Rightarrow$ $x = O(3x + 7)$ for $x > 0$

$\therefore$ $3x + 7 = \Theta(x)$ for $x > 7$.

c) Show $\lfloor x + 1/2 \rfloor = \Theta(x)$.

1. $\lfloor x + 1/2 \rfloor \le x + 1$ $\Rightarrow$ $\lfloor x + 1/2 \rfloor = O(x)$ for $x > 1$
2. $x - 1 \le \lfloor x + 1/2 \rfloor$ since $x - 1 \le \lfloor x + 1/2 \rfloor \le x + 1$ $\Rightarrow x = O(\lfloor x + 1/2 \rfloor)$ for $x > 1$

$\therefore$ $\lfloor x + 1/2 \rfloor = \Theta(x)$ for $x > 1$.

e) Show $log_{10}\ x = \Theta(log_2\ x)$.

1. $log_{10}\ x = \dfrac{\log x}{\log 10} = O(log\ x)$ for $x > 1$
2. $log_2\ x = \dfrac{\log x}{\log 2} = O(log\ x)$ for $x > 1$

$\therefore$ $log_{10}\ x = \Theta(log_2\ x)$ for $x > 1$.

Section 3.3

4- To find x^{2^k} by successive squaring requires $k+1$ multiplications.
To find x^{2^k} by multiplying x by itself requires 2^k multiplications.

The first approach is naturally more efficient.

8- a) Evaluating $3x^2 + x + 1$ at $x = 2$:
Input: $c = 2$, $a_0 = 1$, $a_1 = 1$, $a_2 = 3$.

0. $y = 3$	initialization of y
1. $i = 1, y = 3 \cdot 2 + 1 = 7$	first iteration in *for* loop
2. $i = 2, y = 7 \cdot 2 + 1 = 15$	second iteration in *for* loop

Final Answer: $y = 15$

b) In each iteration of the *for* loop, one multiplication and one addition are used, so the total number of multiplications and additions in the algorithm is n multiplications and n additions.

12- a) To find the maximum of a sequence of n integers:
always requires n-1 comparisons; traverse list from 2 to n, comparing at each iteration

b) To locate an element in a list of n terms with linear search:
Least number of comparisons is 1; occurs if the element to be located is the first element in the list

c) To locate an element in a list of n terms with binary search:
Least number of comparisons is 1; occurs if the element to be located is the first element inspected, i.e. the middle element in the list

16- The search algorithm resembles binary search, except that it divides the list of numbers into 4 sublists. Since binary search has worst-case time complexity of O($\log n$) where $\log n = \log_2 n$, then this search algorithm will worst-case time complexity of O($\log_4 n$) = O($\log n$). This is caused by the fact that the list is continuously subdivided into 4 equal-sized sublists.