Singzon, Ryan
260397455

COMP 250: Introduction to Computer Science
Assignment 2
Fall 2013

Question 1
a) An input of an array sorted in decreasing order will yield the worst possible running time for any fixed size *n*. This is because the value of the next element in the array will always be smaller than the value at the current index, so the conditional statement will always be true, and thus, a swap will occur for every value in the array.

b) T(n) for the BubbleSort algorithm
T_{assign}
$T_{compare} + T_{conditional}$
T_{assign}
$T_{compare} + T_{conditional}$
$T_{compare} + T_{conditional}$
$T_{index} + T_{assign}$
$T_{index} + T_{index} + T_{assign}$
$T_{index} + T_{assign}$
$T_{arith} + T_{assign}$
$T_{arith} + T_{assign}$

Outside loop:
T_{assign}

Inside 1st loop:
$n(T_{compare} + T_{conditional} + 2T_{assign} + T_{index})$

Inside 2nd loop:
$n^2 \text{ x } (2T_{compare} + 2T_{conditional} + 4T_{index} + 4T_{assign} + T_{arith})$

Assuming that primitive operations have roughly the same run time:
$T(n) = T + n(5T) + n^2(13T)$

c) The big-oh representation for this algorithm is $O(n^2)$

Question 2
algo1(n): O(log n)
algo2(n): O(n)
algo3(n): O(1)
algo4(n): $O(n^2)$

260397455
SINGZON, RYAN

FALL 2013
COMP 250
ASSIGNMENT 2

QUESTION 3

Prove that $5n+3\log(n)$ is $O(n)$

Find a constant c such that $f(n) \leq c \cdot g(n)$ for all $n \geq n_0$

$f(n) = 5n + 3\log(n)$

$g(n) = n$

$5n+3\log(n) \leq c \cdot n$ ← choose $c=6$

$5n+3\log(n) \leq 6n$

$3\log(n) \leq n$ This holds for any $n \geq 0$, so choose $n_0 = 0$

QUESTION 4

Prove that $(n+10)^{1.5} + 10n + 3$ is $O(n^{1.5})$

$(n+10)^{1.5} + 10n + 3 \leq c \cdot n^{1.5}$

$(n+10)^{1.5} < (n+n)^{1.5}$ when $n > 10$, so choose $n_0 = 11$

$(n+10)^{1.5} + 10n + 3 \leq (n+n)^{1.5} + 10n^{1.5} + 3n^{1.5}$

$(n+10)^{1.5} + 10n + 3 \leq 15n^{1.5}$

$$\boxed{\begin{array}{c} c = 15 \\ n_0 = 11 \end{array}}$$

SINGIZON, RYAN
260397455

Question 5

Prove that 3^n is not $O(2^n)$

If 3^n was $O(2^n)$, then there would exist some c and n_0 such that $3^n \leq c \cdot 2^n$ for all $n > n_0$.

$$3^n \leq c \cdot 2^n$$
$$\frac{3^n}{2^n} \leq c$$
$$\left(\tfrac{3}{2}\right)^n \leq c$$
$$n \leq \log_{1.5} c$$

→ Since n must be smaller than $\log_{1.5} c$, then the conditions for 3^n to be $O(2^n)$ cannot be satisfied.

Question 6

Find two non-negative functions f and g such that $f(n)$ is not $O(g(n))$ and $g(n)$ is not $O(f(n))$.

In order for f to not be $O(g(n))$ and g to be not $O(f(n))$, they must continually get larger than the other function, only to have the other function become larger at some point afterward, oscillating ~~...~~ forever.

One such example of a pair of functions is:

$f(n) = n$

$g(n) = n^{1+\cos n}$

As n increases, $1+\cos n$ will oscillate between 0 and 2, so depending on the value of n, $g(n)$ will always change from being larger to smaller than n.