NAME: SHOUZAB KHAN
BLAZERID: SKHAN6

Theory Assignment 1

CS 452/652/752 Advanced Algorithms and Applications
Total points: 100

Question 1 Multiple Choice Questions – 10 Points
There could be more than one correct answer; select all correct answers.

Question 1.1 - 2 points

The run-time function T (n) is for representing:
1. the real running time of an algorithm implementation
2. the estimated running time of an algorithm implementation
3. the number of basic instruction performance by an algorithm

Question 1.2 - 2 points

If we are sorting n integers each integer is represented using 32 bits. What is the proper way to define the problem size:
1. 1
2. 32
3. n
4. n^2

Question 1.3 - 2 points

If we are doing bitwise addition between n integers of k bits (k is not constant). What is the proper way to define the problem size:
1. k
2. 32
3. n
4. nk

Question 1.4 - 2 points

The meaning of $f(n) = O(g(n))$ is

1. f(n) and g(n) are the same run time function
2. f(n) and g(n) must have the same growth rate
3. f(n) has the same or lower growth rate than g(n)
4. f(n) belongs to the function set O(g(n))

Question 1.5 - 2 points

If $f(n) = O(g(n))$, $f(n) = \omega(h(n))$, there growth rate should be order as:

1. $f(n) > h(n) <= g(n)$
2. $h(n) > f(n) <= g(n)$
3. $h(n) < f(n) <= g(n)$
4. $g(n) < f(n) <= h(n)$

Question 2 - 10 points

What is the smallest value of n such that an algorithm whose running time is $5000n^2$ runs faster than an algorithm whose running time is 1.5^n on the same machine?

n	$5000n^2$	1.5n
35	6125000	1456109.606
36	6480000	2184164.409
37	6845000	3276246.614
38	7220000	4914369.92
39	7605000	7371554.881
40	8000000	11057332.32
41	8405000	16585998.48
42	8820000	24878997.72
43	9245000	37318496.58
44	9680000	55977744.87
45	10125000	83966617.31

Answer: n = 40 is the smallest value

Question 3 - 10 points

Suppose we are comparing implementations of insertion sort and merge sort on the same machine. For inputs of size n, insertion sort runs in $0.5n^2$ steps, while merge sort runs in 256n log n steps. For which values of n does insertion sort take more steps than merge sort?

N	$0.5n^2$	256nlogn	Algorithm's performance
6478	20982242	20997149	merge sort takes more steps
6479	20988721	21000760	merge sort takes more steps
6480	20995200	21004370	merge sort takes more steps

6481	21001681	21007981	merge sort takes more steps
6482	21008162	21011592	merge sort takes more steps
6483	**21014645**	**21015203**	**insertion sort takes more steps**
6484	21021128	21018814	insertion sort takes more steps
6485	21027613	21022425	insertion sort takes more steps
6486	21034098	21026036	insertion sort takes more steps
6487	21040585	21029647	insertion sort takes more steps

Answer: when $n \geq 6484$

Question 4 - 10 points

Express the following function in terms of Big-O notation:

1. $(n)+50\times n^2 +50$

$T(n) = (n) + 50n2 + 50$

For Big O notation raise all the terms to the highest degree

$T(n) \leq (50+50+1)n^2$

$T(n) \leq 101\ n^2$ for all $n_o \geq 1$

This shows that $c=101$, $n_o = 1$

Therefore $T(n) = O(n^2)$

2. $(n^2)+50\times n^2 +21n^2$

$T(n) = (n2) + 50\ n^2 + 21\ n^2$

For Big O notation raise all the terms to the highest degree

$T(n) \leq (50+21+1)n^2$

$T(n) \leq 72n^2$

$T(n) \leq 72\ n^2$ for all $n_o \geq 1$

This shows that $c=72$, $n_o = 1$

Therefore $T(n) = O(n^2)$

3. $(n^3)–10×n^2$ +500

$T(n) = (n^3) - 10\ n^2 + 500$

For Big O notation raise all the terms to the highest degree

$T(n) \leq (1+|-10|+500)n^3$

$T(n) \leq 511n^3$

$T(n) \leq 511\ n^3$ for all $n_o \geq 1$

This shows that c=511, n_o =1

Therefore $T(n) = O(n^3)$

4. $n^a + n^b$ ($a > b$ and $b > 0$)

T(n) = na + nb

We know that

(a>b,b>0)

For Big O notation raise all the terms to the highest degree

$T(n) \leq n^a + n^a$

$T(n) \leq 2n^a$

Therefore $T(n) = O(n^a)$

Question 5 - 10 points

Given $T(n) = 6n^4 + 5n^3 + 6n^2 + 2n + 99$. Prove that $T(n) = O(n^4)$.

$T(n) = 6(n^4) +5(n^3) + 6n^2 +2n +99$

For Big O notation raise all the terms to the highest degree

$T(n) \leq (6+5+6+2+99)n^4$

$T(n) \leq 118n^4$

$T(n) \leq 118\ n^4$ for all no ≥ 1

We have shown that $T(n)=6n^4+5n^3+6n^2+2n+ 99$ is bounded by $118n^4$ for $n\geq$ so ,

Therefore $T(n) = O(n^4)$

<u>Question 6 - 10 points</u>

Given T (n) = $4n^3 + n^2$. Prove that T (n) = $\Theta(n^3)$.

$T(n) = 4 n^3 + n^2$

$4 n^3 \leq T(n) \leq 4 n^3 + n^2$

Raise the values to highest degree

$4 n^3 \leq T(n) \leq 5 n^3$

Here $c_1 = 4$, $c_2 = 5$ and $n_0 = 1$

Since we have shown both:

$T(n)=O(n^3)$ with $c_2=5$ and $n^0=1$

$T(n)=\Omega(n3)T(n)=\Omega(n^3)$ with $c_1=4$ and $n_0=1$

Therefore $T(n) = \Theta(n^3$

<u>Question 7 - 10 points</u>

Let f(n) and g(n) be two run time functions. State true or false for each of the following statements.

1. $f(n) = O(g(n))$ implies $f(n) = o(g(n))$.

<u>Answer: False</u>

2. $f(n)+g(n)=\Theta(\max\{f(n),g(n)\})$.

<u>Answer: True</u>

3. $f(n) = \Theta(g(n))$ implies $2^{f(n)} = \Theta(2^{g(n)})$.

<u>Answer: True</u>

4. $f(n) = \Theta(f(3n))$.

<u>Answer: True</u>

<u>Question 8 - 30 points</u>

Rank the following functions by the order of growth rate (that is, list them in a list f1(n), f2(n), f3(n), . . . such that f1(n) = O(f2(n)), f2(n) = O(f3(n)), . . .). Partition your list into equivalent classes (e.g., [f(n), g(n)]) such that f(n) and g(n) are in the same class if and only if f(n) = Θ(g(n)). You must prove your answer (by limit test or other means).

$(\sqrt{3})^{\log_3 n}, \ln \ln n, n^{1/2}, n^3, n\lg^2 n, n^2, n^{\ln \ln n}, \ln^2 n^2, 4^{\lg n}$

The ranking can be done using the Limit test and for that, we will need the functions and their derivatives.

		F(n)	**F'(n)**
polynomial	f1(n)	$(\sqrt{3})\log^3$ =n/2	0.5
Logarithmic	f2(n)	ln (ln (n)	1/ (nln(n))
polynomial	f3(n)	n^3	3n^2
Logarithmic	f4(n)	$n\log^n$	logn+1
polynomial	f5(n)	n^2	2n
polynomial	f6(n)	n^(ln(ln(n))	(n^(ln(ln(n))-1)/ln(n)
Logarithmic	f7(n)	ln^2(n^2)	(2ln(n^2))/n
Polynomial	f8(n)	4^logn = 2n	2
polynomial	F9(n)	N1/2	n-1/2

We know that polynomial functions are slower than logarithmic functions.

<u>Comparing f3 and f5</u>

Using limit test

Where lim n ⍰ ∞ (n^3 /n ^2)

after application of limits c= ∞

$ff3(nn) = \omega\omega(ff5(nn))$

F3 has faster growth rate. f5 < f3

<u>Comparing f1 and f8</u>

Using limit test
Where lim n ⍰ ∞ (n/2 / 2n)
after application of limits c > 0 $f1(n) = \Theta(f8(n))$
this shows that their growth rate is equal.
F8 has faster growth rate. f8 < f1

<u>Comparing f8 and f3</u>
Using limit test
Where lim n ⍰ ∞ (2n / n^3)
after application of limits c= ∞ which shows $f3(n) = \omega(f8(n))$
F3 has faster growth rate. f8 < f3

<u>Comparing f8 and f6</u>
Using limit test
Where lim n ⍰ ∞ (2n / (n^(ln (ln(n)))
after application of limits c= ∞ $f8(n) = \omega(f6(n))$
F8 has faster growth rate. f6 < f8
So for the polynomial functions only the growth order is f6 < f1 < f8 < f5 < f3For logarithmic functions

<u>Comparing f2 and f4</u>
Using limit test
Where lim n ⍰ ∞ (ln (ln (n) / nlogn)
after application of limits c= ∞ $f4(n) = \omega(f2(n))$
F4 has faster growth rate. f2 < f4

<u>Comparing f7 and f4</u>
Using limit test
Where lim n ⍰ ∞ (nlogn) / (ln^2(n^2))
after application of limits c= ∞ $f4(n) = \omega(f7(n))$
F4 has faster growth rate. f7< f4
So for the logarithmic functions only the growth order is f2< f7 < f4

Therefore, the order of the growth will be f2< f7 < f4< f6 < f1 < f8 < f5 < f3.

We can now rank the functions and group them into equivalent classes:

1. $f1(n)=n^3$
2. $f2(n) = n^{\log 2}n$
3. $f3(n) = n^2=4^{\log n}$
4. $f4(n)=n \ln \ln n$
5. $f5(n)=n^{1/2}=(\sqrt{3})^{\log 3n}$
6. $f6(n)=\ln^2 n$
7. $f7(n)=\ln \ln n$

This is the final ranking by growth rate, with equivalent classes grouped.

COLLEGE OF ARTS AND SCIENCES
The University of Alabama at Birmingham

Department of Computer Science

Declaration of Independent Completion

Please include the following declaration together with your signature/name in each of your submissions, e.g. in the beginning of a program or on the cover page of your report:

1) If it's an individual assignment, include the following text for the declaration of independent completion in the cover page of your submission:

```
I Shouzab Khan declare that I have completed this assignment in
accordance with the UAB Academic Integrity Code and the UAB CS
Honor Code.  I have read the UAB Academic Integrity Code and
understand that any breach of the Code may result in severe
penalties.
Student signature/initials: SK
Date: 09/22/2024
```