## CSCI203: Algorithms and Data Structures

### Assignment 1 Solutions

Question 1a and 1b:

	Ma.
	Date
Q1a) i.	
(X-101) 1.	15 1/092/092n constant volve so arouth rate O(1) and a constant volve
3.1	and = (100 h) what: This is a lon-longithmic amounth which is a new condition
7.	10'0: This is a constant volve, so growth vote O(1)  (12) 109210921 = (10g 2h) 109212 : This is a log-logarithmic growth, which is a very sow og, log_ ((ntio)) & log_2(10g2h): This is a log-logarithmic growth, which is a very sow
	The state of the s
— <del>  7.</del>	(10g2n) = 3.10g (10g2n): This is a 10g 10g withing and which is a long of
	growth rate but faster than (log_n) og_2 2.  growth rate but faster than (log_n) og_2.  growth rate but faster than log_log_((n+10))
5.	2 log = 1 con : This is a logarithmic growth, which is fuster than
_	log-logarithmic growth.
6.	$\frac{16n^2 = 4n \text{ ; This is a linear growth, so growth rate : } 0(n)$ $\sum_{k=1}^{n} \frac{n(n+1)}{2} \approx \frac{n^2}{2} : \text{ This is a growth rate : } 0(n)$
7.	$\sum_{k=1}^{n} = \frac{n(n+1)}{2} \approx \frac{n^2}{2}$ : This is a quadratic growth, so growth rate is $O(n^2)$ $\sum_{k=1}^{n} = \frac{n(n+1)}{2} \approx \frac{n^2}{2}$ : This is a quadratic growth, so growth rate is $O(n^2)$ $\sum_{k=1}^{3n} = (2^3)^n = 8^n$ : This is exponential growth, which is proper time all polynomial
8.	$27^{-638}$ $2(3^3)^{1093}$ = $13^3$ : This is a cutic amount (a size the variety of $10^3$ )
9.	23" = (23)" = 8": This is exponential growth, which is toster than all polynomial
	growth rate.
10.	(h-3)! : This is a factorial growth, it is faster than exponential growth.
The C	Super tell = 15 was a way of the super of the call
b) ste	fic int doIt (int n) Essent a some management from the some so
	for i < 1 to 10 do : runs exactly to times which is constant, : cost: O(1)
	for $k \in 1$ to ndo : runs n times for each iteration of the outer loop: . cost: O(n)
	jelinen: Outer loop runs lo times, inner loop runs n times. cost: o(n)
	while icm do : for each iterations of the inner loop, the unile loop runs ocloom)
	while icm do : for each iterations of the inner loop, the unile loop runs o(logn)  times, since the inner loop run in times, the total cost: alrigg  m = (m+j)/2: Same as while loop cost: O(nlogn)
: -	The overall complexity of do It() is $O(1) + O(n) + O(n) + O(n\log n) + O(n\log n) = O(n\log n)$
Sto	tic int my Method (intn) {
- 0.0	i < 1 : execute once : cost : O(1)
	while $(i = n)$ $\leq$ : doubles on every iteration: cost: 0 (log n)
	doI+(n> : called on every iteration of while loop, cost: O(logn) × O(nlogn) = O(nlog
	', ← i × 2: This executes 0 (logn) times, : (ost: 0 (logn)
	}
	Neturn 1; : execute once .: cost: 0(1)
3	your )
7	the same of the sa
. 1	the overall complexity of my Method () is $6(1) + 0(\log n) + 0(\log^2 n) + 0(\log n) + 0(1)$
	= 0 (n log2n) AZONE

## Question 1c:

()	$\alpha n = \frac{n^2}{n} + n \ln n$
200	$f(n) = \frac{n^2}{2} + n \lg n$ $g(n) = Sn(4 + 2^{\lg n})$
	f(n) f n (a(n)) is then note any and a strong mad parting trans
51.	f(n) () (g(n)) if there exist a non-negative integer no and positive real
	constant C such that f(n) < g(n) for all n>0
200	$\frac{h^2}{2} + n \lg n \le c \left( 5n \left( 4 + 2^{\lg n} \right) \right)$
	$= \frac{n^2}{2} + n \lg n \leq C \left( 20n + 5n^2 \right)$
	0\ 2
	Agin) a sont sn2 man as almost word as a state of the state of
re inn	1 + 190 (MAIN 6-7 = 100)
12.10	21 THE PARTY OF STREET AND A STREET OF THE CONTROL OF THE STREET OF THE
	The rent = 2 of rother names of the case o
	$=\frac{15}{10}$ When $n=\infty$ , $\frac{19n}{n}=0$ and $\frac{20}{n}=0$
. 19.	to (n-a) : This is a factorial arouth it is fault than expensibilities
	Short f(n) = 1 when h = 00 there with a much of 1 - 1 - 11 of 11 of
	Short f(a) = 1 when h = 00 there with a much of 1 - 1 - 11 - 11 - 11 - 11 - 11 - 11
	Since f(n) = 10 when n = 00 there exist a constant (= 1 or (> 1 such that f(n) < or (g(n)) for all sufficiently large n therefore f(n) & O(g(n))
	Short f(a) = 1 when h = 00 there with a much of 1 - 1 - 11 - 11 - 11 - 11 - 11 - 11
	Since $\frac{f(n)}{g(n)} = \frac{1}{10}$ when $n = \infty$ , there exist a constant (=1 or (>1 such that $f(n) \leq e(g(n))$ for all sufficiently large $n$ , therefore $f(n) \in O(g(n))$ so final answer is yes, $\frac{n^2}{2} + n \lg n \in O(5n(4+2^{\lg n}))$
(+3) 0 : 500 20 : 51	Since $\frac{f(n)}{g(n)} = \frac{1}{10}$ when $n = \infty$ , there exist a constant (=1 or (>1 such that $f(n) \leq \alpha(g(n))$ for all sufficiently (arge n, therefore $f(n) \in O(g(n))$ ).  So final answer is yes, $\frac{n^2}{2} + n \lg n \in O(sn(u+2^{\lg n}))$
(+3) 20 : 20 20 : 21	Since $\frac{f(n)}{g(n)} = \frac{1}{10}$ when $n = \infty$ , there exist a constant (=1 or (>1 such that $f(n) \leq \alpha(g(n))$ for all sufficiently (arge n, therefore $f(n) \in O(g(n))$ ).  So final answer is yes, $\frac{n^2}{2} + n \lg n \in O(sn(u+2^{\lg n}))$
(+3) 0 : 500 20 : 51	Since $\frac{f(n)}{g(n)} = \frac{1}{10}$ when $n = \infty$ , there exist a constant (=1 or (>1 such that $f(n) \leq e(g(n))$ for all sufficiently large $n$ , therefore $f(n) \in O(g(n))$ so final answer is yes, $\frac{n^2}{2} + n \lg n \in O(5n(4+2^{\lg n}))$
(1) 984: 66 94: 96 84: 66 84: 66	Since $\frac{f(n)}{g(n)} = \frac{1}{10}$ when $n = \infty$ , there exist a constant (=1 or (>1 such that $f(n) \leq \alpha(g(n))$ for all sufficiently (arge n, therefore $f(n) \in O(g(n))$ ).  So final answer is yes, $\frac{n^2}{2} + n \lg n \in O(sn(u+2^{\lg n}))$
(1) 984: 66 94: 96 84: 66 84: 66	Since $\frac{f(n)}{g(n)} = \frac{1}{10}$ when $n = \infty$ , there exist a constant (=1 or ( $\geq$ 1 such that $f(n) \leq \alpha(g(n))$ for all sufficiently (arge n, therefore $f(n) \in O(g(n))$ ).  So final answer is yes, $\frac{n^2}{2} + n \lg n \in O(sn(u+2^{\lg n}))$
(1) 984: 66 94: 96 84: 66 84: 66	Since $\frac{f(n)}{g(n)} = \frac{1}{10}$ when $n = \infty$ , there exist a constant (=1 or ( $\geq$ 1 such that $f(n) \leq \alpha(g(n))$ for all sufficiently (arge n, therefore $f(n) \in O(g(n))$ ).  So final answer is yes, $\frac{n^2}{2} + n \lg n \in O(sn(u+2^{\lg n}))$
(1) 984: 66 94: 96 84: 66 84: 66	Since $\frac{f(n)}{g(n)} = \frac{1}{10}$ when $n = \infty$ , there exist a constant (=1 or (\ge 1) such that $f(n) \leq \alpha(g(n))$ for all sufficiently (arge n, therefore $f(n) \in O(g(n))$ ).  So final answer is yes, $\frac{n^2}{2} + n \lg n \in O(5n \cdot (4 + 2^{\lg n}))$
(1) 984: 66 94: 96 84: 66 84: 66	Since $\frac{f(n)}{g(n)} = \frac{1}{10}$ when $n = \infty$ , there exist a constant (=1 or (\ge 1) such that $f(n) \leq o(g(n))$ for all sufficiently (arge n, therefore $f(n) \in O(g(n))$ ).  So fival answer is yes, $\frac{n^2}{2} + n \lg n \in O(sn(4+2^{\lg n}))$
(4) (4) (4) (4) (4)	Since $\frac{f(n)}{g(n)} = \frac{1}{10}$ when $n = \infty$ there exist a constant (=1 or (\ge)1 such that $f(n) < o(g(n))$ for all sufficiently (arge n, therefore $f(n) \in O(g(n))$ ).  So fival answer is yes, $\frac{n^2}{2} + n \log n \in O(sn(4+2^{19n}))$
(4) (4) (4) (4) (4)	Since $\frac{f(n)}{g(n)} = \frac{1}{10}$ when $n = \infty$ , there exist a constant $(=1 \text{ or } (\geq 1 \text{ such that } f(n) \leq e(g(n)))$ for all sufficiently, large $n$ , therefore $f(n) \in O(g(n))$ .  So final answer is yes, $\frac{n^2}{2} + n \log n \in O(5n \cdot (4 + 2^{19^n}))$
(4) (4) (4) (6) (4) (6) (4) (6) (7) (7)	Since $\frac{f(n)}{g(n)} = \frac{1}{10}$ when $n = \infty$ there exist a constant (=1 or (\ge)1 such that $f(n) < o(g(n))$ for all sufficiently (arge n, therefore $f(n) \in O(g(n))$ ).  So fival answer is yes, $\frac{n^2}{2} + n \log n \in O(sn(4+2^{19n}))$
(4) (4) (4) (4) (4)	Since $\frac{f(n)}{g(n)} = \frac{1}{10}$ when $n = \infty$ there exist a constant $C = 1$ or $C \ge 1$ such that $f(n) \le o(g(n))$ for all sufficiently large $n$ , therefore $f(n) \in O(g(n))$ so fival answer is yes, $\frac{n^2}{2} + n \log n \in O(5n \cdot (4 + 2^{19n}))$
(4) (4) (4) (4)	Since $\frac{f(n)}{g(n)} = \frac{1}{10}$ when $n = \infty$ , there exist a constant $(=1 \text{ or } (\geq 1 \text{ such that } f(n) \leq e(g(n)))$ for all sufficiently, large $n$ , therefore $f(n) \in O(g(n))$ .  So final answer is yes, $\frac{n^2}{2} + n \log n \in O(5n \cdot (4 + 2^{19^n}))$

## Question 2a:

	No.
_	Date
20	function Stan (A 1001 1611)
24)	function Sum (A left, right)
	it left > right: the time apparent component: cost:0(1)
10	return 0: constant operation for base case: cost: O(!)
650	else if left = right: time component: (0st: 0(1) : cost: 0(1)
	return A [1894]: Accessing an array element is constant time operation : cost: 0C1
	(Nid = Hoor ((18ft/right)/2): Arithemic and floor operation are constant time operation
	Isum = Sum (A, 18ft, mid) ]: This is recursive call, the problem size is recursive call: (0st: 0(3))
	naturn Isum + rsum + A[mid]: adding three value is a constaint - time opera
	:. (ost:0(1)
	(dypolypol x "45" x) (o (n) T. T
	:. recurrance relation is $T(n) = 2T(\frac{n}{2}) + C$
	5 D ( n log 109 m)
	expanding the recoverance relation: $T(n) = 2T(\frac{n}{2}) + C$
	==12[2](22)76]76
	= 227 ( 1/2 ) +3C HE OF SE
100	$T(\overline{2}^2) = 2^2[2T(\overline{2}^3) + c] + 3C$
	$= 2^3 T \left(\frac{1}{2^3}\right) + 4C + 3C$
	Generalised the recurrence lebotion: T(n) = 2 t T (3 x) + (2 t - 1) (
	The recursive call will stop when the base condition is met, which is $\frac{h}{2} k = 1$
	M 1 = 1 = 1 = 1 = 1 = 1 = 1 = 1 = 1 = 1
	$V = 2^{\kappa}$
	K = Id N
	c how wo 1 into neneralised recurrence Relation: T(n)=2kT(2k)+(2k-1)(
	= 2 ((790)+(20-1))
	=nT(\frac{n}{n}) + (n-1) \c
	T(1)=1 = = nT(1) + cn-C
	an = nn + cn - C
	:- the running time efficiency is O(n) -
	The worst case asymptotic complexity of this algorithm is the same as the algorithm
	that linearly iterates through the list. :. The worst-case asymptotic complexity is O(n)
	The best-case asymptotic complexity of this algorithm & only reduces the constants
-	associated with non-recursive work, but does not fundamentally change the reccurrence.  The best-case asymptotic complexity is O(n)  A'ZONE

# Question 2b and 3 part i, ii, iii:

ь) (	he best-cuse inputs and worst-cuse inputs are the same becomes the algori
a	ilways diven divides the input into halves recursively until the base cases a
P	HI VI elements of the input are processed regardless of their volues or an
a	and there is no condition to stop recursion early based on input prope
7500	meann Alighet : Accepting an army element is endered that
ison der	When it has common roof his swampy: (2) ( High Hall) your & but
3)()	$T(n) = 4T(\frac{n}{4}) + \frac{n}{\log_{10}}, \text{ and } T(1) = 1$
19. : 17	a=4, b=4, c=1, p=-1 (/mor 1+bim 1)mig=min
914	adjust from + From + Ainst: ablan 1-= q apr 1 = air = ==
:	. T(n) = 0 (n logs x logs logs n)
	= 0 (n 109 x 109 4
	= 0 (n 10gu 10gun)
-34	DECEDITION THE recommence relation: T(n) = DITCEDITION SHE parlimones
(i)	I(n) = 2T (n) + In, and T(1) = 1
	a=2, b=4, c=\frac{1}{2}, p=0
	$a=2$ , $b=4$ , $c=\frac{1}{2}$ , $p=0$
	2E+3++(37) T'S ≈
- 1	$T(n) = O(n^{\log_2 \alpha} \times \log_2 P^{+1}(n))$ $= O(n^{\log_2 \alpha} \times \log_4 (n))$
12.4	the O(n ("ga x logg in) and ant pools one that the system of sail
	= 0 (n = 10g un) x
	£ = N
(111)	$T(n) = 2T(\frac{n}{2}) + n \lg^2 n  and  T(1) = 1$
20	a=2+ b=2 (c=1 mp=2 merrier believing orn) of profition !
3-19	$\frac{a}{b} = \frac{2}{3} = 1$ , where $p = 2$
36.43	
	$T(n) = O(n^{100/2^2} \times 109/2^3 n)$
	= 0 (n log <sub>2</sub> <sup>3</sup> n)
	- THE REMOVED AND SHOP IN MENT
	The worst case requipitite missexity of the elements the serve or the con-
10	that have promise through the last with the source of a disposition on plant is
184	The next 200 regenerate complement of the displacement and making the continuence
CITE )	TAMES NOT COMPANY PROPERTY AND ADDRESS OF THE PARTY OF TH

## Question 3 part iv, v, vi:

	No
	Date
1) , ,	
(v) T(n) = 2T(=)+nigh, and T(+)=1	
$a=2$ , $b=2$ , $C=1$ , $P=1$ $\frac{a}{b} = \frac{2}{2} = 1$ , $p=1$	
6 = 2 = 1 p=1	
- T/D> - 6 ( - 100 La . P+1 )	
( 100 3 (M)	
= 0 (n <sup>3</sup>   00 2 n)	
$T(n) = 0 \left( n^{109} b^{4} \log_{b} P^{+1} n \right)$ $= 0 \left( n^{109} 2^{2} \log_{2}^{2} n \right)$ $= 0 \left( N \log_{2}^{2} n \right)$	
$v)$ $T(n) = 2T(\frac{h}{2}) + h^3$ and $T(1) = 1$	
A=2, $b=2$ , $C=3$ , $P=0$	
$\frac{Q_2}{Q_2} = \frac{Q_2}{Q_2} = $	
:. T(n) = 6(nc x log, Pn)	
$= \frac{0(n^3 \times 10^9 \times 0^9)}{0.00}$	
$=0(n^3)_{\star}$	
VI) T(n)=T(n-3)+3N+3	
= T(n-1-1) + 3(n-1) + 3 + 3n + 3	
= T(n-2) + 3(n-1) + 3 + 3n + 3 — 0	
T(n)=T(n-1-2)+3(n-1-1)+3+3(n-1)+1 +3+3n+	-3
- T(h-3)+3 (h-2) +3+3(n+)+3+3n+3 -	<b>—</b> Ø
T(n) = T(n-k) + 3[(n-k+1)+(n-k+2)++n] + 3k	
Since base case T(1)=1, for k=n-1	
T(n) = T(1) + 3[1+2+3++n] + 3(n-1)	-1)
n(nti)	
= T(1)+3 (n(n+1)) + 3(n-1)	
$= 1 + \frac{3n+3}{2} + 3(n-1)$	
$\frac{1+30^2+30+30-3}{2}$	
$=\frac{3n^2+3n}{2}+3n-2$	
$=\frac{3}{2}n^2+\frac{3}{2}+3n-2$	
:- dominant term = 3 n2	
$T(n) = O(n^2) x.$	

#### Question 4a:

#### Screenshot of the output

```
Ciprogram Files(WindowsAp × + v - D × 123 - 456 0 0 0 Invalid input: String contains invalid characters or format.

Press Enter to terminate the program.
```

### Question 4b:

```
# Recur for the rest of the string
return recursive_convert(param[1:], result) # Recursive calls
```

The algorithm processes one character of the string in each recursive call. For a string of length n, there are n recursive calls to process all characters.

Each recursive call performs constant work (character conversion and basic arithmetic operations).

```
digit = ord(param[0]) - ord('0') # Convert character to integer
result = result * 10 + digit
```

The ord(param[0]) - ord('0') and the arithmetic operations are  $\Theta(1)$ .

Since there are n recursive calls and each call does  $\Theta(1)$  work, the total time complexity is  $\Theta(n)$