CSc 225 Assignment 2:
Runtime Analysis and Proofs

Due date:

The submission deadline is 11:55pm on Monday, June 1st, 2020.

How to hand it in:

Submit your assignment2.pdf file through the Assignment 1 link on the CSC225 ConneX page.
IMPORTANT: the file submitted **must** have a **.pdf** extension.

Exercises:

1. Determine the number of assignment (A) and comparison (C) operations in the following:

   a. **Algorithm** Q1a(n)

$result \leftarrow 1$	1A
for $i \leftarrow 1$ to n^2 **do**	1A + n^2(1C + 1A) + 1C
for $j \leftarrow 1$ to i **do**	n^2(1A + $\frac{(n^2+1)}{2}$(1C + 1A) + 1C)
$temp \leftarrow i + j$	$\frac{(n^2(n^2+1))}{2}$(1A + 1A)
$result \leftarrow result + temp$	
end	
end	
return $result$	1A

$4 + n^2(4) + \frac{(n^2(n^2+1))}{2}(4) = 4 + 4n^2 + \frac{(n^4+n^2)}{2}(4) = 4 + 4n^2 + 2n^4 + 2n^2 = 2n^4 + 6n^2 + 4$

   b. **Algorithm** Q1b(n)

$result \leftarrow 1$	1A
for $i \leftarrow 1$ to n **do**	1A + n(1C + 1A) + 1C
$j \leftarrow 1$	n(1A)
while $j \le n$ **do**	X(1C) + n(1C)
$result \leftarrow result + 1$	X(1A)
$j \leftarrow j * 2$	X(1A)
end	
end	
return $result$	1A

HINT: Assume n is a power of 2

$4 + n(4) + n \log n(3) = 3n \log n + 4n + 4$

Note: while-loop repeating $logn + 1$ times also correct

What is X?

Let's look at the values of j:

$1, 2, 4, 8, 16, 32, \dots, n/2, n.$

This can also be written as:

$2^0, 2^1, 2^2, 2^3, 2^4, \dots, 2^{i-1}, 2^i$, where i is the number of times j is doubled, and also the number of iterations of the loop

So how many iterations are there?

$2^i = n$, solving for i: $\log n = i$

2. Solve the following recurrence relations, given an integer $n \geq 1$, through substitution:

a. $T(n) = \begin{cases} 1, & n = 1 \\ 2T(n-1) + 1, & n \geq 2 \end{cases}$

$$T(n) = 2T(n-1) + 1$$
$$T(n-1) = 2T(n-2) + 1$$
$$T(n-2) = 2T(n-3) + 1$$
$$\ldots$$
$$T(2) = 2T(1) + 1$$
$$T(1) = 1$$

$$T(n) = 2T(n-1) + 1$$
$$T(n) = 2(2T(n-2) + 1) + 1 = 4T(n-2) + 3$$
$$T(n) = 4(2T(n-3) + 1) + 3 = 8T(n-3) + 7$$
$$T(n) = 8(2T(n-4) + 1) + 7 = 16T(n-4) + 15$$
$$= 2^4 T(n-4) + 2^4 - 1$$
$$\ldots$$
$$T(n) = 2^i T(n-i) + 2^i - 1$$

Let $i = n - 1$ to simplify term to $T(1)$, we get:

$$\ldots$$
$$T(n) = 2^{(n-1)} T\big(n - (n-1)\big) + 2^{(n-1)} - 1$$
$$= 2^{(n-1)} T(1) + 2^{(n-1)} - 1$$
$$= 2^{(n-1)} + 2^{(n-1)} - 1 = 2(2^{n-1}) - 1$$
$$= 2^n - 1$$

b. $T(n) = \begin{cases} 4, & n = 1 \\ T(n-1) + 2n, & n \geq 2 \end{cases}$

$$T(n) = T(n-1) + 2n$$
$$T(n-1) = T(n-2) + 2(n-1)$$
$$T(n-2) = T(n-3) + 2(n-2)$$
$$\ldots$$
$$T(2) = T(1) + 2(2)$$
$$T(1) = 4$$

$$T(n) = T(n-1) + 2n$$
$$T(n) = \big(T(n-2) + 2(n-1)\big) + 2n$$
$$= T(n-2) + 2n + 2(n-1)$$
$$T(n) = (T(n-3) + 2n) + 2n + 2(n-1)$$
$$= T(n-3) + 2n + 2(n-1) + 2(n-2)$$
$$\ldots$$
$$T(n) = T(n-i) + 2n + 2(n-1) + \cdots + 2(n-(i-1))$$

Let $i = n - 1$ to simplify term to $T(1)$, we get:

$$T(n) = T\big(n - (n-1)\big) + 2n + 2(n-1) + \cdots 2(n - ((n-1) - 1))$$
$$T(n) = T(1) + 2n + 2(n-1) + \cdots 2(n - n + 2))$$
$$T(n) = T(1) + 2n + 2(n-1) + \cdots 2(2))$$
$$T(n) = 4 + 2(n + (n-1) + (n-2) + \cdots + 3 + 2)$$
$$T(n) = 4 + 2(\frac{n(n+1)}{2} - 1)$$
$$T(n) = 4 + n^2 + n - 2$$
$$T(n) = n^2 + n + 2$$

We know the sum from 1 to n can be expressed as $\frac{n(n+1)}{2}$ and this is the sum from 2 to n, so: $\frac{n(n+1)}{2} - 1$

3. Proof by induction

a. Prove the following statement using induction:

$$\sum_{i=1}^{n}(i)*(i!) = (n+1)!-1, \text{ for all } n \geq 1.$$

Base case:
LHS: $1*1! = 1$, RHS: $(1+1)!-1 = (2)!-1 = 2-1 = 1$. So LHS = RHS = 1

Inductive Hypothesis:
Assume the statement holds for some $k < n$.
So, assume that $\sum_{i=1}^{k}(i)*(i!) = (k+1)!-1$

Inductive Step:
Show that it holds for $k+1$. So, show that: $\sum_{i=1}^{k+1}(i)*(i!) = (k+2)!-1$
LHS:
$\sum_{i=1}^{k+1}(i)*(i!) = \sum_{i=1}^{k}(i)*(i!) + (k+1)*(k+1)!$, then, if we substitute from I.H. we get:
$= (k+1)!-1+(k+1)*(k+1)!$ factoring out (k+1)!
$= (k+1)!\,(1+k+1)-1 = (k+1)!\,(k+2)-1 = (k+2)!-1$ = RHS

So by principle of induction $\sum_{i=1}^{n}(i)*(i!) = (n+1)!-1$, is true for all $k \geq 1$

b. Prove the following statement using induction:

$$1+6+11+16+\cdots+(5n-4) = n(5n-3)/2, \text{ for any positive integer } n \geq 1.$$

Base case:
LHS: 1
RHS: 1(5(1)-3)/2 = 2/2 = 1

Inductive Step:
Inductive Hypothesis:
Assume the statement holds for some $k < n$.
So, assume that $1+6+11+16+\cdots+(5k-4) = k(5k-3)/2$

Inductive Step:
Show that it holds for $k+1$. So, show that:

$$1+6+11+16+\cdots+(5k-4)+(5(k+1)-4) = \frac{(k+1)(5(k+1)-3)}{2}$$

LHS:
$1+6+11+16+\cdots+(5k-4)+(5(k+1)-4)$, then, if we substitute from I.H. we get:

$$= \frac{k(5k-3)}{2}+(5(k+1)-4) = \frac{k(5k-3)}{2}+\frac{2(5(k+1)-4)}{2}$$
$$= \frac{5k^2-3k}{2}+\frac{10k+2}{2}$$
$$= \frac{5k^2+7k+2}{2} = \frac{(k+1)(5k+2)}{2}$$
$$= \frac{(k+1)(5(k+1)-3)}{2}$$
$= $ RHS

So by principle of induction, $1+6+11+16+\cdots+(5n-4) = n(5n-3)/2$, for all $k \geq 1$

4. Use a *loop invariant* to prove that the algorithm below **returns** a^n.

Algorithm Q4(a, n)
Input: Positive integers $a \geq 0$ and $n \geq 0$
Output: a^n

$i \leftarrow 1$
$pow \leftarrow 1$
while $i \leq n$ **do**
$pow \leftarrow pow * a$
$i \leftarrow i + 1$
end
return pow

Loop invariant: $pow = a^{i-1}$ and $i \leq n$

Base case:
Before the loop is entered $pow \leftarrow 1$ and $i \leftarrow 1$.
LHS $= pow = 1$
RHS $= a^{1-1} = a^0 = 1 =$ LHS. Thus, the invariant holds.

Inductive Hypothesis:
Assume the invariant holds up to some iteration $i = k$, where $k \leq n$.
So, assume that $pow = a^{k-1}$ and $k \leq n$ at the start of iteration k

Inductive Step:
Show that it holds for the next iteration of the loop, when $i = k + 1$.
So, show that $pow = a^k$ at the start of iteration $k + 1$.

By the I.H. at the *start* of iteration k, $pow = a^{k-1}$
During iteration k, we have $pow = pow * a$
$= a^{k-1} * a$ (by substitution from IH)
$= a^k$ at the end of iteration k
Thus, at the start of iteration $k + 1$, $pow = a^k$ and k increments to $k + 1$. The invariant holds

Termination:
The final iteration is when $k = n$.
At the end of that iteration, i increments to $n + 1$ and $pow = a^n$ and the loop condition evaluated to false (the loop is not entered again). Thus, the value returned is a^n

5. Prove each of the following using the definition of Big-Oh. You must provide constants c and n_0 to satisfy the definition of Big-Oh as defined in class.

a. $(n+2)^4$ is $O(n^4)$

$$(n+2)^4 = n^4 + 4n^3 + 6n^2 + 4n + 1$$
$$\le n^4 + 4n^4 + 6n^4 + 4n^4 + n^4 \text{ for all } n >= 1$$
$$\le 16n^4$$

$\therefore\ (n+2)^4 \in O(n^4)$ with $c = 16$ and $n_0 = 1$

b. $\frac{n^4+n^2+1}{n^3+1}$ is $O(n)$

$$\frac{n^4+n^2+1}{n^3+1} = \frac{n^4}{n^3+1} + \frac{n^2}{n^3+1} + \frac{1}{n^3+1}$$
$$\le \frac{n^4}{n^3} + \frac{n^2}{n^3} + \frac{1}{n^3}$$
$$\le n + n + n \text{ for all } n >= 1$$
$$\le 3n$$

$\therefore\ \frac{n^4+n^2+1}{n^3+1} \in O(n)$ with $c = 3$ and $n_0 = 1$

c. Find the smallest integer x such that $5n^3 + 3n^2 logn$ is $O(n^x)$

Can't be 2 as $\lim_{n\to\infty} \frac{cn^2}{n^3} = 0$ for any integer constant c

Try $x = 3$: $5n^3 + 3n^2 logn \le 5n^3 + 3n^3$ for all $n \ge 1$

$$\le 8n^3$$

$\therefore\ 5n^3 + 3n^2 logn \in O(n^3)$ with $c = 8$ and $n_0 = 1$.

Thus, $x = 3$ is the smallest integer, and $5n^3 + 3n^2 logn$ is $O(n^3)$

6. Order the following functions by their big-Oh notation. Group together (for example, by underlining) functions that are big-Theta of one another (no justification needed).
Note: $\log n = \log_2 n$ unless otherwise stated.

$5n \log n$	2^{40}	$\log \log n$	4^n	$14n$	$2^{\log n}$
$(n \log n)^2$	$\sqrt{\log n}$	$n \log_5 n$	n^3	$25n^{0.5}$	$7n^{5/2}$
$1/n$	5^{n^2}	$n^2 \log n$	$\sqrt{n}$	$n^{0.01}$	2^{2^n}

$1/n$, 2^{40}, $\log \log n$, $\sqrt{\log n}$, $n^{0.01}$, $\underline{\sqrt{n}, \quad 25n^{0.5}}$, $\underline{2^{\log n}, \quad 14n}$,
$\underline{5n \log n, \quad n \log_5 n}$, $n^2 \log n$, $(n \log n)^2$, $7n^{5/2}$, n^3, 4^n, 5^{n^2} 2^{2^n}