Assignment 1.1
Asymptotic Analysis of Iterative Algorithms
Submission due: 01/November/2023

1. Let f(n) and g(n) be asymptotically positive functions. Prove or disprove each of the following conjectures.

   a) f(n) = O(g(n)) implies g(n) = O(f(n)).
   b) f(n) + g(n) = Θ(min(f(n),g(n)).
   c) f(n) = O(g(n)) implies 2f(n) = O(2g(n))
   d) f(n) = O(g(n)) implies g(n) = Ω(f(n))
   e) f(n) = Θ(f(n/2))

2. Is $2^{n+1} = O(2^n)$? Is $2^{2n} = O(2^n)$? $2^{n+1} \rightarrow 2^n \times 2 \rightarrow O(2^n)$
   $2^{2n} \rightarrow (2^n)^2 \,!= O(2^n)$

3. In the light of topics covered so far, the long and short of algorithm analysis is:

   - Don't count the number of operations. How many operations you execute within or outside the loop doesn't matter. The number of times you loop, that's what matters.

   - For example, if T(n) = an + b, a represents the number of operations within the loop, b denotes the number of operations outside the loop, n denotes the number of times the loop executes. Since a and b are constants and ignored, T(n) = Θ(n). What survives for determining time complexity is the number of times the loop runs. i.e. n.

Based on this knowledge, determine the time complexity T(n) for the below scenarios.

Sl.No	Loop	T(n)	Justification, if any
1	for (i=1; i<=n; i++)	$O(n)$	loop work n time
2	for (i=1; i<=n; i=i+2)	$O(n/2)$	i=i+2
3	for (i=1; i<=n; i=i*2)	$O(\log_2 n)$	i=i*2
4	for (i=1; i<=n; i++) for (j=1; j<=n; j++)	$O(n^2)$	nested loop
5	for (i=1; i<=n; i++) for (j=1; j<=i; j++)	$O(n^2)$ 2	n 1+2+3...n
6	for (i=1; i<=n; i++) for (j=1; j<=n; j=j*2)	$O(n \times \log_2 n)$	
7	for (i=1; i<=n; i++) for (j=1; j<=n; j++)		

	for (k=1; k<=n; k++)	$O(n^2+n)$	
8	for (i=1; i<=m; i++) for (j=1; j<=n; j++)	$O(m \times n)$	

9	for (i=1; i<=n; i++)
	
	for (k=1; k<=m; k++)
		$O(n+m)$	

10	for (i=1; i<=n; i++)
	      for (j=1; j<=n; j++)
	for (k=1; k<=m; k++)
		$O(n^2+m)$	

4. Algorithm A uses 10nlogn operations, while algorithm B uses n^2 operations. Determine the value n_0 such that A is better than B for n >= n_0.

5. Order the following in the increasing order of growth. Group together those functions of the same order.
   6nlogn, 2^{100}, log log n, $\log^2 n$, $2^{\log n}$, 4^n, sqrt(n), $n^{0.01}$, 1/n, $4n^{3/2}$, $3n^{0.5}$, 5n, $2n\log^2 n$, 2^n, $n\log_4 n$, 4n, n^3, $n^2\log n$, $4^{\log n}$, sqrt(log n), n!, n^3

6. Show that $10n^2 + 5n + 1 \in \Omega(n)$ by finding appropriate c and n_0.

7. Show that $10n^2 + 5n + 1 \in \theta(n^2)$ by finding appropriate c and n_0.

8. Is $10n^2 + 5n + 1 \in \theta(n^3)$? Justify your answer.

9. Prove the following:
   a) $5n^3 + n^2 + 6n + 2 = O(n^3)$
   b) $5n^3 + n^2 + 6n + 2 = O(n^4)$
   c) $5n^3 + n^2 + 6n + 2 = \Omega(n^2)$
   d) $5n^3 + n^2 + 6n + 2 = \Theta(n^3)$

10. Suppose that each row of an n x n array A consists of 1's and 0's such that, in any row of A, all the 1's come before any 0's in that row. Provide a
    a. $O(n^2)$ algorithm for finding the row of A that contains most 1's.
    b. O(n) algorithm for the same.

A note for you

Following table shows you the importance of designing good algorithms. Table shows algorithms' with various complexities actual running time for various input sizes.

	n	$n \log_2 n$	n^2	n^3	1.5^n	2^n	$n!$
$n = 10$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	4 sec
$n = 30$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	18 min	10^{25} years
$n = 50$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	11 min	36 years	very long
$n = 100$	< 1 sec	< 1 sec	< 1 sec	1 sec	12,892 years	10^{17} years	very long
$n = 1{,}000$	< 1 sec	< 1 sec	1 sec	18 min	very long	very long	very long
$n = 10{,}000$	< 1 sec	< 1 sec	2 min	12 days	very long	very long	very long
$n = 100{,}000$	< 1 sec	2 sec	3 hours	32 years	very long	very long	very long
$n = 1{,}000{,}000$	1 sec	20 sec	12 days	31,710 years	very long	very long	very long

1) a) $f(n) = O(g(n))$ implies $g(n) = O(f(n))$

eg:- $f(n) = n$ & $g(n) = n^2$

$f(n)$ is $O(g(n))$ but

$g(n)$ is not $O(f(n))$

$\therefore$, disapproved.

b) Its true

eg:. $f(n) = n^2$ $g(n) = n$

either one of the function dominates

hence their sum will also grow.

c) True.

Multiplying with the same constant on both sides have same effect.

d) $g(n)$ is upper bound of $f(n)$ that

means $f(n)$ is lower bound of $g(n)$

eg $f(n) = n^2$ $g(n) = n^2$

e) The given statement holds true if n is constant. But if n is not a constant then then f(n) grows faster than f(n/2) and ∴ the given condition doesn't hold.

4) A - $10n\log n$ B - n^2 $n \geq n_0$

$10n\log n < n^2$

$10\log n < n$

$10\log n_0 = n_0$ → At threshold

$\log(10) = 1$ ∴ $10\log(10) = 10$

ie $10\log(n_0) = n_0$

∴, $n_0 = 10$

5) $1 \ln < \log(\log(n)) < \log^2(n) < \sqrt{\log(n)}$
$< n^{0.001} < 3n^{0.5} < n^{3/2} < \sqrt{n} < n < n\log n$
$< n^2(\log n) < n^3 < 4n < 6n\log n < 2n\log^2(n) <$
$2^{\log n} < 4^{\log n} < 2^{100} < n! < 4^n$

6) $\cancel{10}n^2 \cancel{+} \cancel{5n} \cancel{+} \cancel{1} \geq c.n$

$n^2 \geq c.n$

$n^2 \geq c.n$

Let $c = 1$ and $n = 5$

$25 \geq 5$ hence proved.

7) $c_1.n^2 \leq 10n^2 + \cancel{5n} \cancel{+1} \leq c_2.n^2$

Case 1:

$c_1 n^2 \leq 10n^2$

Let $c_1 = 1$ then $n^2 \leq 10n^2$

$\forall n \geq 1$

Case 2:

$$10n^2 \leq C_2 n^2$$

Let $C_2 = 100$ then $10n^2 \leq 100n^2$
$\forall\ n \geq 1$

$\therefore$, $n_0 = 1$, $C_1 = 10$ and $C_2 = 100$.

8) $C_1 . n^3 \leq f(n) \leq C_2 . n^3$ is false when $f(n) = 10n^2 + 5n + 1$ as the lower bound won't satisfy.

9)

a) $n^3 \geq n^3$ $\Rightarrow$ $n^3 = n^3$ $\therefore$ proved

b) $n^3 \leq n^4$ eg: $8 < 16$
$\Rightarrow 2^3 < 2^4$ $\therefore$, proved.

c) $n^3 \geq n^2$ $8 > 4$
$\therefore$, proved

d) $C_1 . n^3 \leq 5n^3 \leq C_2 n^3$

Let $C_1 = 2$ and $C_2 = 10$.

$\Rightarrow 2n^3 \leq 5n^3 \leq 10n^3$

$\therefore$, proved.

10)

a)