1 With the pseudo-code below, determine the final value of sum when n=23.

Algorithm 1

```
1: procedure
2: sum = 0
3: for (i=2, i<n, i=i+3) do
4: if i mod 2 == 0
5: sum = sum + i
```

```
2 - 0 + 2 = 2

5

6 - 2 + 8 - 10

11

14 - 10 + 14 = 24

17

20 - 4 24 + 20 = 44

18
```

- 2 Use the pseudo-code below to answer the following question.
 - (a) Without working through the pseudo-code, how many times with val be updated in Line 7?
 - (b) What is the final value of val.

Algorithm 2

```
1: procedure
2: dFour = [1, 2, 3, 4]
3: dSix = [1, 2, 3, 4, 5, 6]
4: val = 0
5: for (x in dFour) do
6: for (y in dSix) do
7: val = val + x·y
```

Wednesday, April 26, 2023 8:38 PM

> 3 Using the pseudo-code below, what is the ending value for count when n = 100? When n = 10,000? When $n = 10^{42}$?

Algorithm 3

```
1: procedure
     count=0
3:
     while (n>1) do
       count++
4:
        n = n/10
```

$$N_1 = 100 = 10^2$$
 \rightarrow county = 2
 $n_2 = 10,000 = 10^4$ \rightarrow county = 41
 $n_3 = 10^{42}$ \rightarrow county = 42

- 4 This question assumes familiarity with modular arithmetic discussed in Unit 2. In the pseudo-code below, Line 7 performs string addition (concatenation), for example:
 - "a" + "b" = "ab". Even if the addends look like numbers, it will still do string addition:
 - "2" + "3" = "23". Answer the questions below.
 - a. What is the output of the algorithm when x=99 and b=2?
 - b. What is the output of the algorithm when x=1642 and b=8?

Algorithm 4

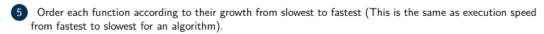
```
1: procedure Expand(x, b)
     Input: integers x and b
     Output: outVal, x written in base b
3:
4:
    outVal = an empty string
6:
    while (x>0) do
7:
       outVal = string(x mod b) + outVal
       x = x div b
     return(outVal)
```

```
a) x=aa b=2
    · (99 div2 = 49, 99 mod2 = 1) -> (49 div2 = 24,49 mod2 = 1) -> (24 div2 = 12 24 mod2 = 0) ->
(10001) > (10001) > (10001) > (10001) > (10001) > (10001)
        = 1100011
```

Page 3

Wednesday, April 26, 2023

9:09 PM



- (a) $n \cdot \log n$
- (b) $\log_{10} n$
- (c) 10^n
- (d) n^{10}
- (e) 10n
- (f) 100
- (g) n^2

6 The provided expressions are the processing time of an algorithm for problems of size n. For each expression, find the *lowest possible* Big- \mathcal{O} complexity stated in simplest form.

- (a) $100n^2 + 0.0002n^3 + 10,000$
- (b) $20n + 2n \log_{10} n + 30$
- (c) $\log_{10} 5 + \log_2 7$
- (d) $5n + \sqrt{n}$
- (e) $0.01(2^n) + n^5 + \ln n$
- (f) $100 \cdot 4^n + 200 \cdot 3^n$

7 $f(x) = 6x \log_4 x$ is which of the following? Mark all that apply.

 $\bigcirc \ \mathcal{O}(6) \quad \bigcirc \ \mathcal{O}(\log x) \quad \bigcirc \ \mathcal{O}(6\log x) \quad \bigcirc \ \mathcal{O}(x\log x) \quad \bigcirc \ \mathcal{O}(x^2) \quad \bigcirc \ \mathcal{O}(2^x)$

Since Big. 0 is apperbound,
$$J(x) \in O(x\log x) \subset O(x^2) \cap O(2^x)$$
, so $O(x\log x) \cap O(x^2) \cap O(2^x)$

8 Let $h(x) = 6x^4 + 2x^3 + x + 100$. Which of the following is true? Mark all that apply.

$$\bigcirc \ h(x) \in \mathcal{O}(x^4) \quad \bigcirc \ h(x) \in \Theta(x^4) \quad \bigcirc \ h(x) \in \Omega(x^4)$$

D(x4) upper bound D(x4) lower bound O(x1) +ight bound

Page 4

Thursday, April 27, 2023 11:12 PM

9 Let h(x) = 6 log₂ x. Which of the following is true? Mark all that apply.

- $\bigcirc h(x) \in \mathcal{O}(4x^2)$ $\bigcirc h(x) \in \Theta(4x^2)$ $\bigcirc h(x) \in \Omega(4x^2)$

Glogx is taster than 4x

10 Let $f(x) = x^2 \log_3 x$. Which of the following is true? Mark all that apply.

 $\bigcirc \ f(x) \in \mathcal{O}(2x \log_{10} x) \quad \bigcirc \ f(x) \in \Theta(2x \log_{10} x) \quad \bigcirc \ f(x) \in \Omega(2x \log_{10} x)$

1/ 2> 1+

11 Let $g(x) = 2x \log_{10} x$. Which of the following is true? Mark all that apply.

 $\bigcirc g(x) \in \mathcal{O}(x^2 \log_3 x)$ $\bigcirc g(x) \in \Theta(x^2 \log_3 x)$ $\bigcirc g(x) \in \Omega(x^2 \log_3 x)$

/ Jx cx2

Nove: 0 - upper 2 -> Lower Smaller shower faster

Using the provided pseudo-code, find the worst case performance in Big-O notation.

Algorithm 5 Some more messy pseudocode

- 1: procedure SOMEPROCEDURE2
- 3: for i=0; i<n; i++ do
- j=i+j
 - A. $\mathcal{O}(\log n)$
 - B. $\mathcal{O}(n \log n)$
 - C. $\mathcal{O}(n^2)$

Page 5

Thursday, April 27, 2023 11:33 PM



Using the provided pseudo-code, find the worst case performance in Big- $\mathcal O$ notation.

Algorithm 6 Some messy pseudo-code

- 1: procedure SOMEPROCEDURE for i=1 and i <= n do 3: j=1 while j<n do 4: 5: j=j+2
 - A. $\mathcal{O}(\log n)$
 - B. $\mathcal{O}(n \log n)$
 - C. $\mathcal{O}(n^2)$
 - D. *O*(*n*)





14 Using the provided pseudo-code, find the worst case performance in Big-O notation. Assume we know that someMethod(n) is $O(\log n)$.

Algorithm 7 Some pseudocode with a method in it

- 1: procedure SOMEPROCEDURE3
- for i=0; i<n; i++ do 3:
- j=someMethod(n) 4:
 - A. $\mathcal{O}(\log n)$
 - B. $\mathcal{O}(n \log n)$
 - C. $\mathcal{O}(n^2)$
 - D. O(n)

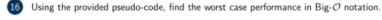


Using the provided pseudo-code, find the worst case performance in Big-O notation.

Algorithm 8 Some more messy pseudocode

- 1: procedure SOMEPROCEDURE4
- while n>1 do
- n=n/2
 - A. $\mathcal{O}(\log n)$
 - B. $\mathcal{O}(n \log n)$
 - C. $\mathcal{O}(n^2)$
 - D. *O*(*n*)





Algorithm 9

- 1: procedure 2: a = 1c = 03: while a < n do4: for i = 0; i < n; i++ do c = c + 16: a = a * 3
 - A. $\mathcal{O}(\log n)$
 - B. $\mathcal{O}(n \log n)$
 - C. $\mathcal{O}(n^2)$
 - D. $\mathcal{O}(n)$



· While loop runs	logan Times
, for look invo v	=

-> nlogn