

**Start time: December 3, at 10AM CT**

**I gave edit access to everyone**

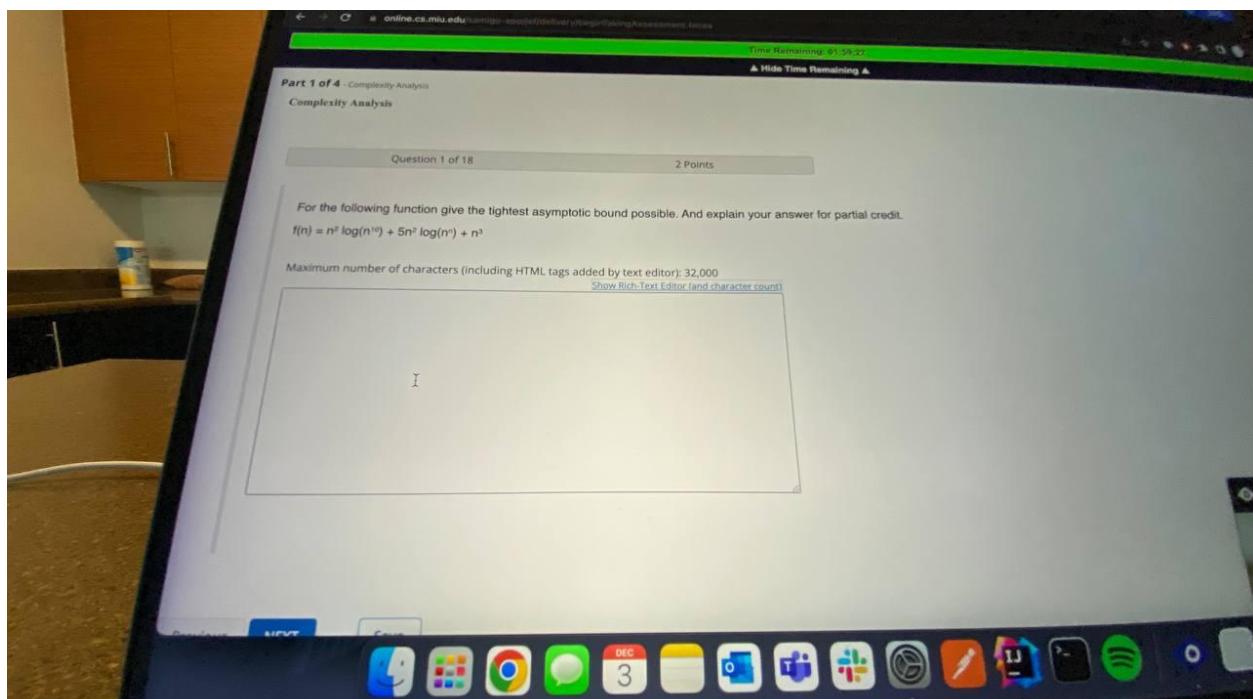
**Please, during the exam, read the answers, and write them to the system with your own words, if you are caught on plagiarism, because you just copy pasted the answer from here, it is going to be your fault! Play with words, and make it different.**

## Questions@Answers

**Answer statuses:** NotStarted, Editing, Done

Please, when you start searching for answer, change status to "Editing", and once wrote down full answer change status to "Done"

### Question 1



**Answer 1 [DONE]**

$$\Rightarrow n^2 \log(n^{10}) + 5n^2 \log(n^n) + n^3$$

$$\Rightarrow 20n^2 \log n + 5n^3 \log(n) + n^3$$

=>  $O(n^3 \log(n))$  Please check if there are errors

Answer B [Done]

$$f(n) = n^2 \log(n^{10}) + 5n^2 \log(n^n) + n^3$$

$$= 10n^2 \log n + 5n^3 \log n + n^3$$

$$10n^2 \log n = O(n^2 \log n)$$

$$5n^3 \log n = O(n^3 \log n)$$

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

The tightest asymptotic bound is  $O(n^3 \log n)$

---

## Question 2

The screenshot shows a digital assignment interface. At the top left is a 'Table of Contents' button. Below it, under 'Part 1 of 4 - Complexity Analysis', is a section titled 'Complexity Analysis'. A horizontal bar indicates 'Question 2 of 18' on the left and '2 Points' on the right. The main question text reads: 'For the following function give the tightest asymptotic bound possible. And explain your answer for partial credit.' Below this is the function definition  $f(n) = 2^{100}n^4 + 2^n$ . A note below the function states: 'Maximum number of characters (including HTML tags added by text editor): 32,000' and includes a link 'Show Rich-Text Editor (and character count)'. A large text input area is provided for the answer.

Answer 2 [DONE]

Since  $O(n^4) < O(2^n)$  then answer is:  $O(2^n)$

---

## Question 3

Give the worst case running time in Big-Oh notation for the following function, **Show how you reached your result:**

```
void foo()
{
    int count=0;
    for(int i=0; i<1000; ++i)
    {
        count += i;
        for(j=0; j<i; ++j)
        {
            System.out.println(j);
        }
        for(k=0; k<i; ++k)
        {
            System.out.println(k);
        }
    }
}
```

Maximum number of characters (including HTML tags added by text editor): 32,000

[Show Rich-Text Editor \(and character count\)](#)

### Answer 3 [Done]

$O(n*(n+n)) \rightarrow O(2*n^2) \rightarrow$  since  $n=1000$  const, the answer is  $O(1)$

---

### Question 4

Give the worst case running time in Big-Oh notation for the following function, **Show how you reached your result:**

```
int foo(int n)
{
    int sum = 0;
    int i = 1;
    int j = n*n;
    while(i++ < j--)
        sum++;
    return sum;
}
```

Maximum number of characters (including HTML tags added by text editor): 32,000

[Show Rich-Text Editor \(and character count\)](#)

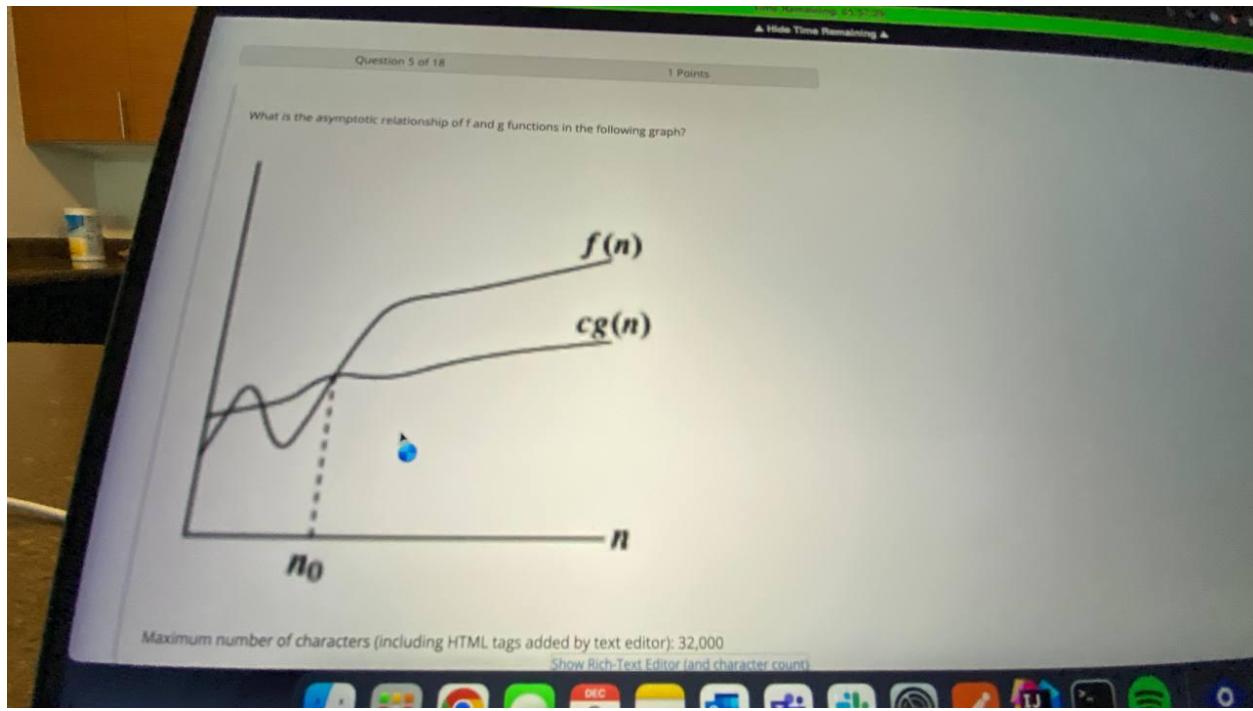
Mi

Answer 4 [Done]

$O(n^2)$

---

Question 5



Answer 5 [done]

---

$$f(n) = \text{Big Omega}(g(n))$$

---

Question 6

**Part 1 of 4** - Complexity Analysis

Complexity Analysis

Question 6 of 18

1 Points

Given that  $f(n) = 4n^2$ , then  $f(n)$  is  $\Omega(n \log n)$ .

True

False

[Reset Selection](#)



[Previous](#)

[NEXT](#)

[Save](#)

Answer 6 [Done]

True.  $f(n) > g(n)$

---

Question 7

**Part 1 of 4** - Complexity Analysis

**Complexity Analysis**

Question 7 of 18

1 Points

Given that  $f(n) = 4n^2$ , then **f(n)** is  $O(n^3)$

True

False

[Reset Selection](#)

[Previous](#)

[NEXT](#)

[Save](#)

Answer 7 [Done]

true

Because  $4n^2$  can be simplified as  $n^2$ . And the time complexity will become  $O(n^2)$ , hence  $f(n) \leq c(n^3)$ , for all  $c > 0$ ,  $n > 0$

---

Question 8

Table of Contents

**Part 1 of 4** - Complexity Analysis

Complexity Analysis

Question 8 of 18

1 Points

Given that  $f(n) = 4n^2$ , then **f(n)** is  $\Theta(n)$ .

- True
- False

[Reset Selection](#)

[Previous](#)

**NEXT**

[Save](#)

Answer 8 [Done]

False

---

Question 9

**Part 2 of 4 - Analysis**

Information you might need.

1. Master Formula

Suppose  $T(n)$  satisfied

$$T(n) = \begin{cases} d & \text{if } n = 1 \\ aT\left(\left\lceil \frac{n}{b} \right\rceil\right) + cn^k & \text{otherwise} \end{cases}$$

Where  $k$  is none negative integer and  $a, b, c, d$  are constants with  $a > 0, b > 1, c > 0, d \geq 0$  then

$$T(n) = \begin{cases} \Theta(n^k) & \text{if } a < b^k \\ \Theta(n^k \log n) & \text{if } a = b^k \\ \Theta(n^{\log_b^a}) & \text{if } a > b^k \end{cases}$$

2.  $x = b^y \implies \log_b x = y$

3.  $\sum_{i=0}^{n-1} i = \frac{n(n-1)}{2}$

8.5

**Part 1 of 4 - Complexity Analysis**

**Complexity Analysis**

Question 9 of 18

1 Points

Sort the given functions f1, f2, f3 and f4 in Increasing order based on asymptotic complexity?

- $f1(n) = n^2$
- $f2(n) = n!$
- $f3(n) = n \log n$
- $f4(n) = 7^n$

Maximum number of characters (including HTML tags added by text editor): 32,000

Show Rich-Text Editor (and character count)

Answer 9 [Done]

Since  $O(n \log n) < O(n^2) < O(7^n) < O(n!)$

Answer is  $f3 < f1 < f4 < f2$

---

Question 10

Question 10 of 18

2 Points

Give tight asymptotic bound for the following recurrence, show your work.

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

Maximum number of characters (including HTML tags added by text editor): 32,000

[Show Rich-Text Editor \(and character count\)](#)

Answer 10 [DONE]

As per description,  $T(n) = 4T(n/2) + n^2$

$a=4$ ,  $b=2$ ,  $k=2$   $b^k = 2^2 = 4$  So,  $a=b^k$  therefor from the master formula we can write  $T(n) = \Theta(n^k \log n) = \Theta(n^2 \log n)$

Alternative answer:

---

Question 11

Time Remaining: 01:54:48  
▲ Hide Time Remaining ▲

2.  $x = b^y \Rightarrow \log_b x = y$

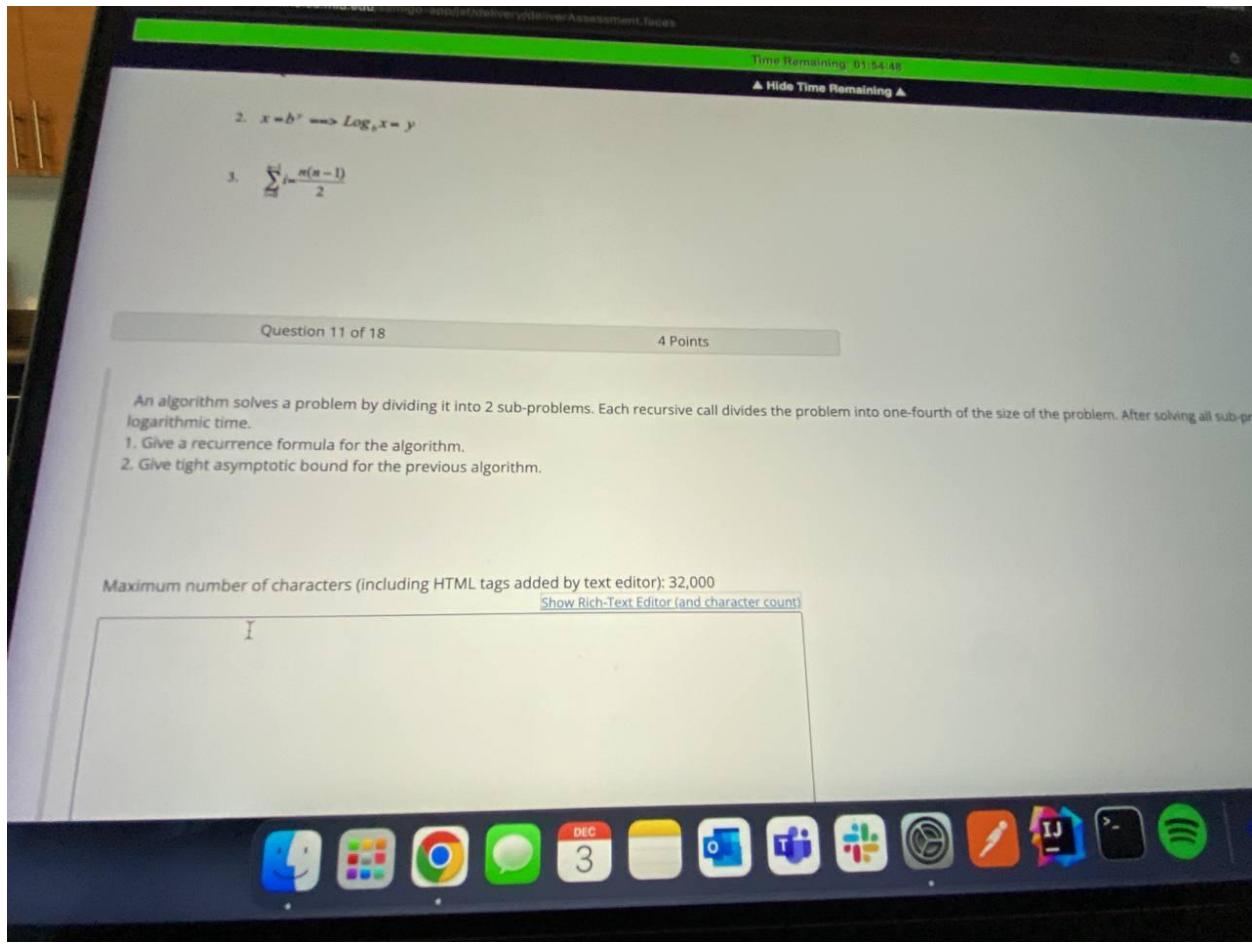
3.  $\sum_{n=1}^{k-1} i = \frac{n(n-1)}{2}$

Question 11 of 18      4 Points

An algorithm solves a problem by dividing it into 2 sub-problems. Each recursive call divides the problem into one-fourth of the size of the problem. After solving all sub-problems, the algorithm takes constant time to combine the results.

1. Give a recurrence formula for the algorithm.  
2. Give tight asymptotic bound for the previous algorithm.

Maximum number of characters (including HTML tags added by text editor): 32,000  
[Show Rich-Text Editor \(and character count\)](#)



### Answer 11 [Done]

1.  $T(n) = 2(T(n/4)+c)$
2.  $a=2 b=4 k=1$   
 $A < b^k$   
 $2 < 4$   
 $\Theta(n^k) = \Theta(n)$

### Optional

1.  $T(n) = 2(T(n/4)+\log n)$
2.  $a=2 b=4 k=0$   
 $A > b^k$   
 $2 > 1$   
 $T(n) = \Theta(n^{\log_4(2)})$

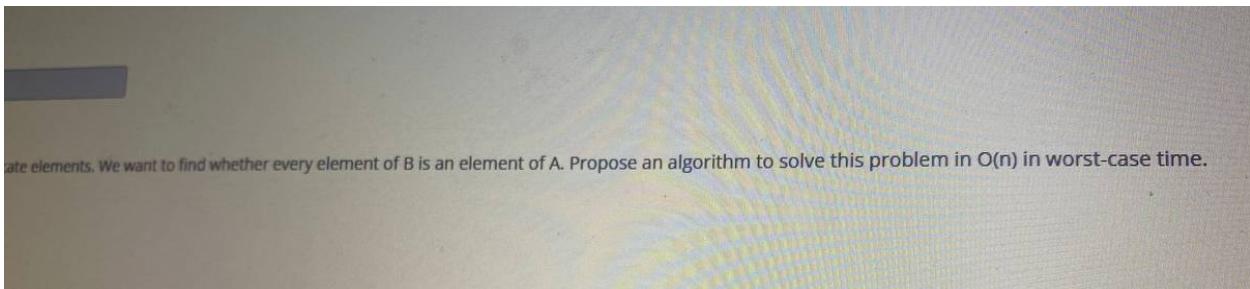
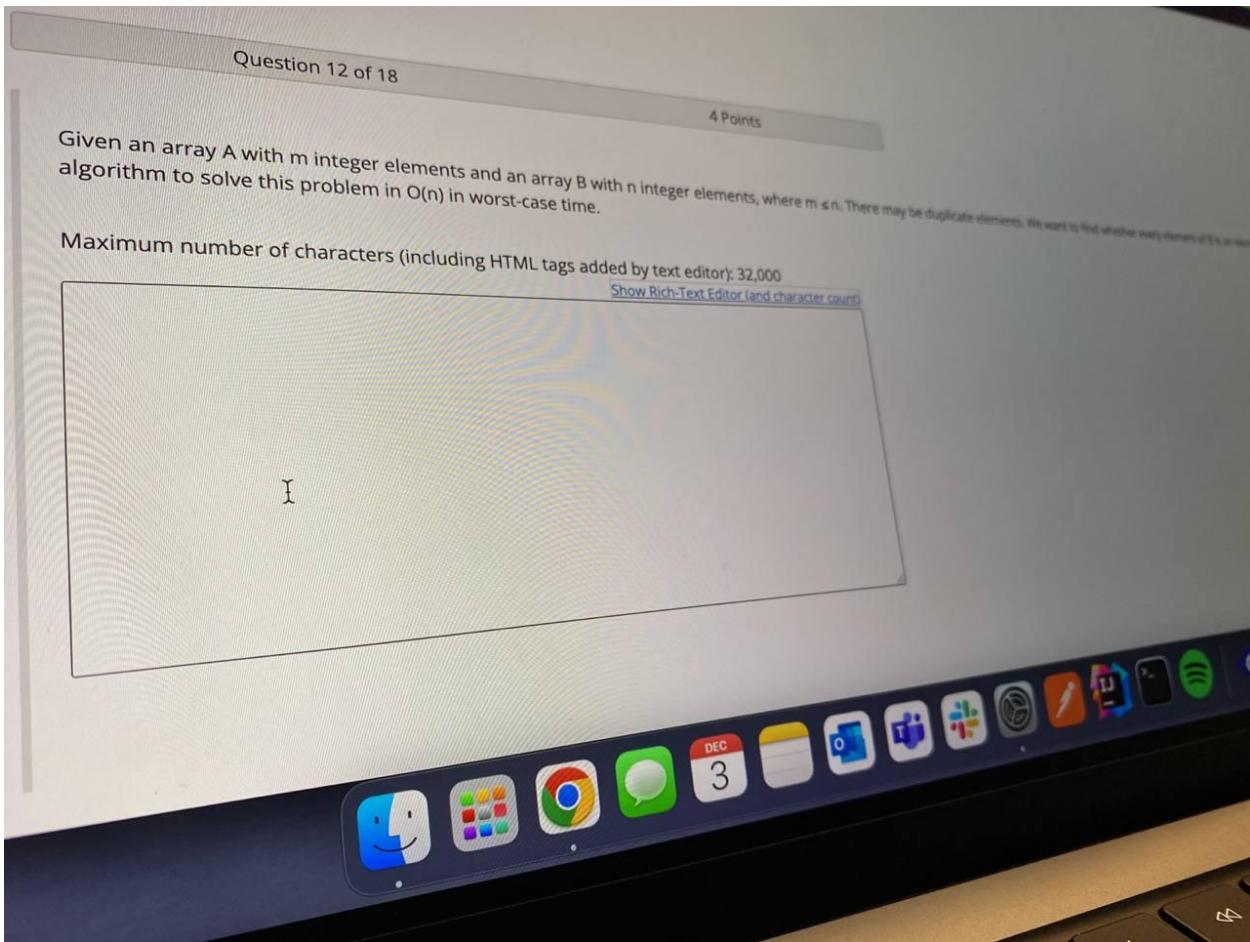
## Question 12

Question 12 of 18      4 Points

Given an array A with m integer elements and an array B with n integer elements, where  $m \leq n$ . There may be duplicate elements. We want to find whether every element of B is an element of A. Propose an algorithm to solve this problem in  $O(n)$  in worst-case time.

Maximum number of characters (including HTML tags added by text editor): 32,000

Show Rich-Text Editor (and character count)



## Answer 12 [Done]

A[m], B[n]

Create a set : lets call it Bset as set uses hashing and contains method runtime is O(1)  
boolean result = true

Loop thorough A and put its elements in a set (Aset)

Loop thorough B {

    If Aset does not contains B[i] then result = false

}

return result

Since  $m \leq n$   $O(n+m) \leq O(2n)$  which is equal to  $O(n)$

Alternative

ALGORITHM isSubset(A, n, B, m)  
Input an Array A and n length of Array  
Input an Array B and m length of Array

```
for i <- 0; i < n; i++ {
    for (j <- 0; j < m; j++)
        if (A[i] == B[j])
            break;
    if (j == m)
        return false;
}
return true;
```

$O(n*m) \rightarrow O(n)$  because  $n > m$

---

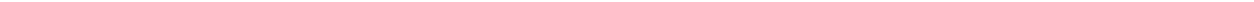
Question 13

Question 13 of 18      2 Points

Use the **asymptotic definition** of Big-O to prove that the order of  $f(n) = n^2 + 5n$  is  $O(n^2)$ .

Maximum number of characters (including HTML tags added by text editor): 32,000

[Show Rich-Text Editor \(and character count\)](#)



Answer 13 [Done - check it]

$$f(n) = n^2 + 5n$$

$$n^2 + 5n \leq c(n^2) - \text{let } n_0 = 1, c \geq 6$$

$$n^2 \leq c(n^2) - 5n$$

Try  $c=6$  and  $n_0=1$

$$1 \leq 6(1) - 5$$

$1 \leq 1$ , which is True.

It will then be true for all  $c \geq 6$ .

Hence  $f(n) = O(n^2)$

---

Question 14

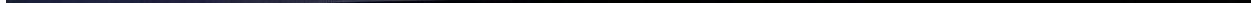
Part 3 of 4 - Sorting algorithms

Question 14 of 18 2 Points

Given that the running time worst case for merge-sort is better than quick-sort. Why quick-sort is commonly used?

Maximum number of characters (including HTML tags added by text editor): 32,000

[Show Rich-Text Editor \(and character count\)](#)



Answer 14 [Done]

1- Auxiliary Space: Quick sort is an in-place sorting algorithm. In-place sorting means no additional storage space is needed to perform sorting. Merge sort on the other hand requires a temporary array to merge the sorted arrays and hence it is not in place.

2- Worst case: The worst case of quicksort  $O(n^2)$  can be avoided by using randomized quicksort. It can be easily avoided with high probability by choosing the right pivot. Obtaining an average case behavior by choosing the right pivot element makes it improvise the performance and become as efficient as Merge sort.

3- Locality of reference: Quicksort in particular exhibits good cache locality and this makes it faster than merge sort in many cases like in a virtual memory environment.

4- Tail recursion: QuickSort is tail recursive while Merge sort is not. A tail recursive function is a function where the recursive call is the last thing executed by the function. The tail recursive functions are considered better than nontail recursive functions as tail-recursion can be

## Question 15

Part 3 of 4 - Sorting algorithms

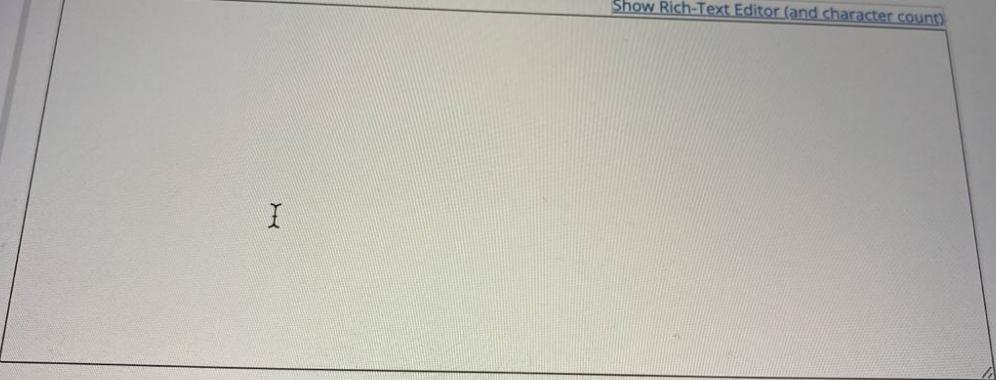
Question 15 of 18 4 Points

Sort the following array using Radix Sort. Show your work.

[10, 135, 199, 87, 358, 230, 25, 15, 22]

Maximum number of characters (including HTML tags added by text editor): 32,000

Show Rich-Text Editor (and character count)



[10, 135, 199, 87, 358, 230, 25, 15, 22]

010|135|199|087|358|230|025|015|022

0	1	2	3	4	5	6	7	8	9
010		022			135		087	358	199
230					025				

0	1	2	3	4	5	6	7	8	9
	010	022	230		358			087	199
	015	025	135						

0	1	2	3	4	5	6	7	8	9
010	135	230	358						
015	199								
022									
025									
087									

010|015|022|025|135|199|230|358

[10, 15, 22, 25, 87, 135, 199, 230, 358]

## Alternative solution

[10, 135, 199, 87, 358, 230, 25, 15, 22]

[ 0, 1, 2 , 3, 4, 5 , , 6, 7, 8, 9]

[[10, 230], [], [22], [], [], [135, 25, 15], [] ,[87] ,[358] ,[199]] // By 0th digit

[10, 230, 22, 135, 25, 15, 87, 358, 199]

[[[], [10,15], [22,25], [230, 135], [], [358],[ ], [ ], [87],[199]] // By 10th Digit

[10, 15, 22, 25, 230, 135, 358, 87, 199]

[[10, 15, 22, 25, 87], [135, 199], [230], [358], [],[],[],[],[],[]] // By 100th digit

=> [10, 15, 22, 25, 87, 135, 199, 230, 358]

Answer 15 [Checked - good]

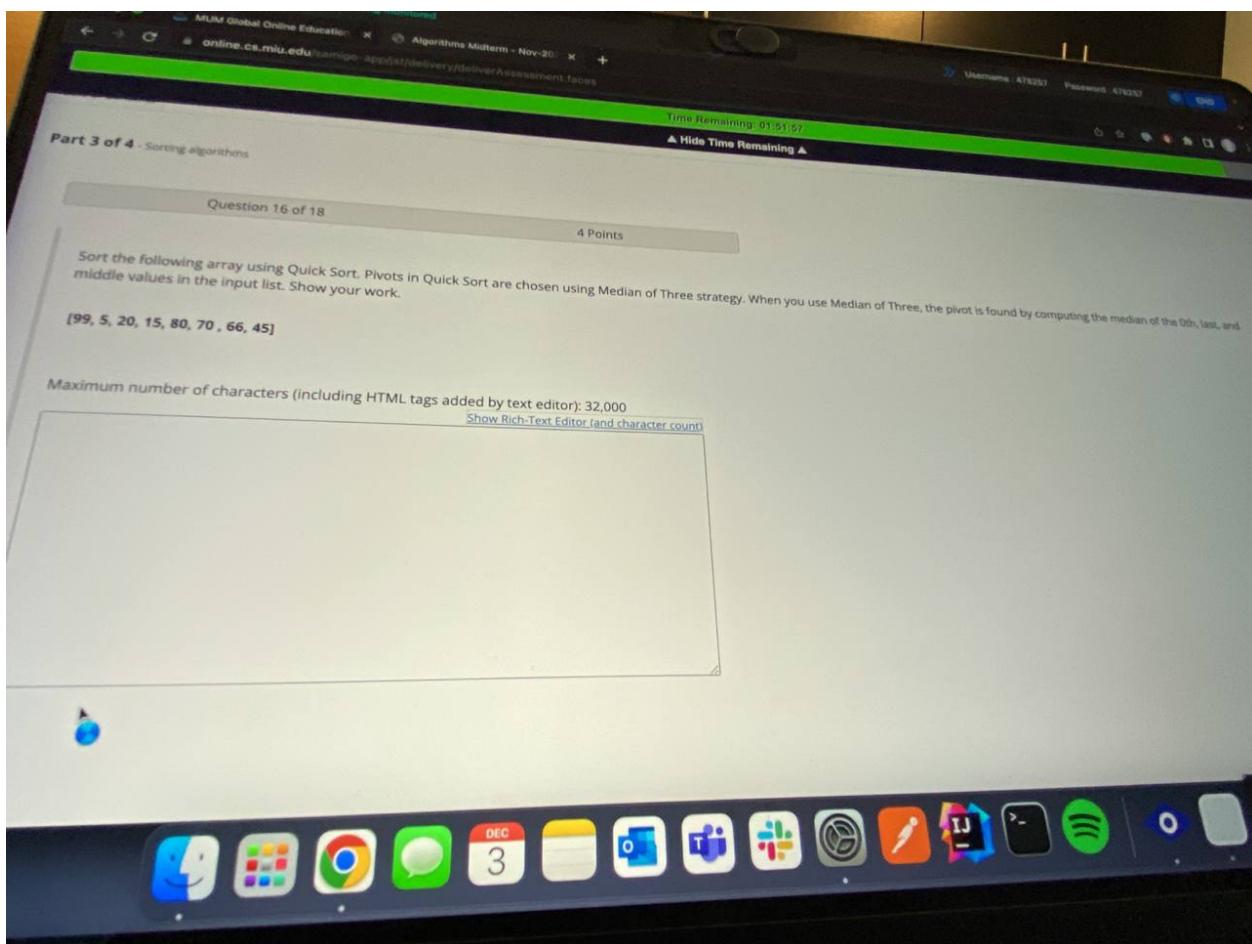
10,230,22,135,25,15,87,358,199

10,15,22,25,230,135,87,358,199

10,15,22,25,87,135,199,230,358

---

Question 16



Answer 16 [Done],

Check this guys (I think it is correct)

[99, 5, 20, 15, 80, 70, 66, 45]

First = 99, Last = 45, Middle = array[floor((0+7)/2)] = array[3] = 15  
**So Pivot= 45 (median of 99, 45, 15)**

Lower = [ 5, 20, 15]      Pivot= 45,      Greater = [99, 80, 70, 66]

---

2.

Sorting the lower [5, 20, 15]

First = 5, Last=15 Middle = 20

**Pivot = 15 (median of 5, 20, 15)**

Lower = [5]    Pivot = 15,    Greater = [20]

Sorting the greater , [99, 80, 70, 66]

First = 99,      Last=66,      Median = array[floor(0+3)/2] = array[1] = 80

**Pivot = 66 (Median of 99, 66, 80)**

Lower = []      Pivot= 66,      Greater =[ 99,80, 70]

---

3.

Sorting Greater for #2 [99, 80, 70]

First= 99,      Last=70,      Median = array[floor(0+3)/2] = array[1] = 80

**So pivot = 70 , Median (99, 70, 80)**

Lower=[],    Pivot = 70,    Greater=[99, 80]

---

4

Sorting the greater , [99, 80]

**Choosing either as a pivot,**

L = [99],    Pivot = 80,    G=[]

—  
Finally merging all

[80, 99]

[70, 80, 99]

[66, 70, 80, 99]    and [5, 15, 20]

[5, 15, 20, 66, 70, 80, 99]

---

### Question 17

Part 3 of 4 - Sorting algorithms

Question 17 of 18

4 Points

Sort the following using Merge Sort based on the integers (first row). No need to write the algorithm. Show all your work for partial credit.

4	8	4	1	2	2	3	8
Z	Y	A	H	Q	B	X	C

Maximum number of characters (including HTML tags added by text editor): 32,000

Show Rich-Text Editor (and character count)

I

Answer 17 [Done]

Merge Sort is a Divide and Conquer sorting algorithm

Partition phase:

[4 8 4 1]

[2 2 3 8]

[4 8] [4 1] [2 2] [3 8]

[4] [8] [4] [1] [2] [2] [3] [8]

Recursive call, base case & Merge

[4 8] [1 4] [2 2] [3 8]

[1 4 4 8] [2 2 3 8]

Sorted result: [1 2 2 3 4 4 8 8]

---

## Question 18

Part 4 of 4 - (5 points) SCI Question

Question 18 of 18

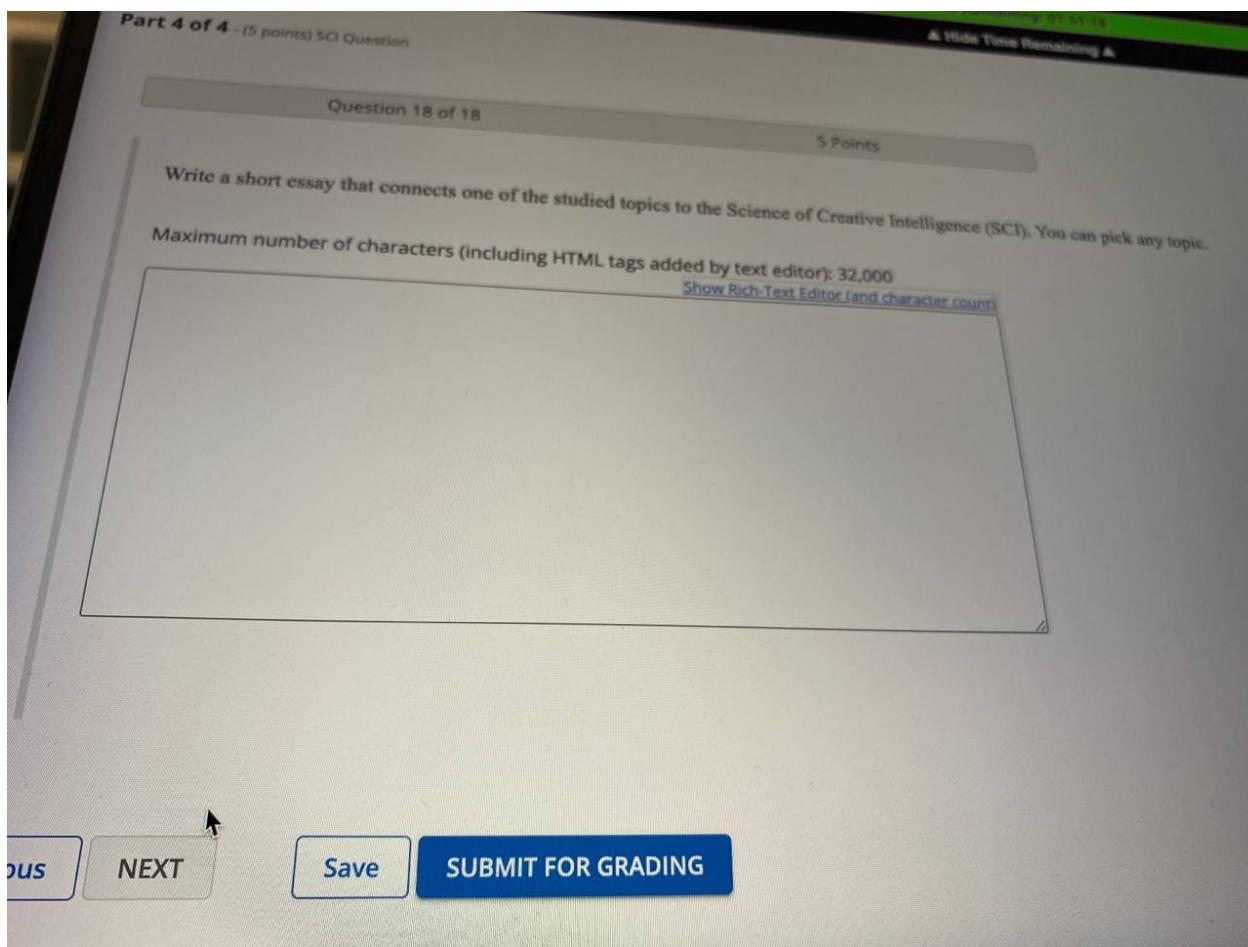
5 Points

Write a short essay that connects one of the studied topics to the Science of Creative Intelligence (SCI). You can pick any topic.

Maximum number of characters (including HTML tags added by text editor): 32,000

Show Rich-Text Editor (and character count)

[PREVIOUS](#) [NEXT](#) [Save](#) [SUBMIT FOR GRADING](#)



Answer 18 [NotStarted]

---

---

