First Midterm Exam

- Due Mar 7 at 11:59pm
- Points 80
- Questions 6
- Available Mar 3 at 8am - Mar 7 at 11:59pm
- Time Limit None

Instructions

Please answer the following questions. Always look for the **most efficient** algorithm /the **tightest bound**.

No makeup exam option is available.

No deadline extension option is available.

As it appears in the course syllabus, students should not discuss the questions with others and are not allowed to get help from the internet, friends, or any other resources to answer the questions. You are expected to turn in the results of your own effort (not the results of a friend's efforts). Even when not explicitly asked, you are supposed to justify your answers concisely.

The exam is individual.

One Attempt ONLY. No Exception is Considered. Ensure that you do NOT hit the Submit button unless if you are done with the exam.

As you answer the questions, your answers will be saved automatically (as shown in the screenshot below), and you can come back and review or finish the rest of the questions later.

sample-1.jpg

Again: You will submit the answers ONLY once you are done with all the questions and/or plan to complete the test.

If you click on submit sooner, whatever you have solved by that time will be considered your answers and graded.

You have no time limit (as long as you submit before the deadline, Friday, 03/07/25, BEFORE 11:59 pm). Please note the link to take the exam disappears sharp at 11:59 pm on Friday.

This quiz was locked Mar 7 at 11:59pm.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	91 minutes	80 out of 80

Correct answers are no longer available.

Score for this quiz: 80 out of 80
Submitted Mar 7 at 2:30pm
This attempt took 91 minutes.

Question 1
15 / 15 pts

A) **(10 points)** Write a recursive function in **C++** or **Java** that reverses a given array of **n** integers **(10 points)**.

Example for an odd number of elements:

Input: [1, 20, 3, 14, 5]
Output: [5, 14, 3, 20, 1]

Example for an even number of elements:

Input: [10, 20, 30, 40]
Output: [40, 30, 20, 10]

Note 1: Do not use any loops. The function should reverse the array without using any **extra array storage** other than the input array.

Note 2: Ensure you write a recursive function. No points for an iterative version of the function.

B)) Analyze the time complexity of your recursive function by **first** (1) **expressing it as a recurrence relation (2.5 points)**.

Then, (2) solve the recurrence relation and express the time complexity in Big-O notation, making sure it is of order **O(n). (2.5 points)**

Your Answer:

// Sol_A : Note : only wrote a function to reverse the array (using recursion)

```
void revArr(int arr[], int l, int r)
{
   if (l >= r)
      return;

   int temp = arr[l];
   arr[l] = arr[r];
   arr[r] = temp;

   revArr(arr, l + 1, r - 1);
}
```

// Sol_B :

1) Recurrence relation: T(n) = T(n-2) + O(1)

2) By expanding terms, T(n) = T(n-2) + O(1)
T(n-2) = T(n-4) + O(1)
T(n-4) = T(n-6) + O(1)
....
T(2) = T(0) + O(1)

after k steps, T(n) = T(n-2k) + O(k) ....eq(1)
when n-2k <= 0, we reach the base case T(0) = O(1),
so, n - 2k = 0
k = n/2,

By substituting this value in eq(1) T(n) = O(n/2) = O(n)
Final Complexity: O(n)

Question 2
15 / 15 pts

Apply the Master Theorem to the following recurrence relations and express each in Big-O notation. **Show all your work clearly and completely.** Assume T(1)=1 in all cases.

Each problem is worth **5 points**:

- **3 points** for demonstrating the step-by-step solution.
- **2 points** for providing the correct final answer.

A) $T(n)= 3\ T(n/3) + 45$

B) $T(n)= 9\ T(n/3) + 2n^2$

C) $T(n)= 4\ T(n/2)+ O(n^3)$

Your Answer:

Master Theorem is used to solve recurrence relations of the form: **$T(n) = aT(n/b) + f(n)$**

A) $T(n)= 3\ T(n/3) + 45$:

1 . $a = 3$, $b= 3$, $f(n) = 45$

2. $\log_b a = \log_3 3 = 1$

3. compare $f(n)$ with $n^{\log_b a}$: $f(n) = 45 = O(1)$ and $n^{\log_b a} = n$

4. In this case $O(n^{1-e})$ for $e =1$, $f(n) = O(1)$, this case falls under case 1 of the master theorem

5. Solution is $T(n) = \boxed{\Theta}(n^{\log_b a}) = \boxed{\Theta}(n)$

So, $T(n) = O(n)$

B) $T(n)= 9\ T(n/3) + 2n^2$:

1 . $a = 9$, $b= 3$, $f(n) = 2n^2$

2. $\log_b a = \log_3 9 = 2$

3. compare $f(n)$ with $n^{\log_b a}$: $f(n) = 2n^2 = O(n^2)$ and $n^{\log_b a} = n^2$

4. In this case $f(n) = O(n^2) = O(n^{\log_b a})$, falls under case 2 of the master theorem

5. Solution is $T(n) = \boxed{\Theta}(n^{\log_b a} \log(n)) = \boxed{\Theta}(n^2\log(n))$

So, $T(n) = O(n^2\log(n))$

C) $T(n)= 4\ T(n/2)+ O(n^3)$:

1 . $a = 4$, $b= 2$, $f(n) = O(n^3)$

2. $\log_b a = \log_2 4 = 2$

3. compare $f(n)$ with $n^{\log_b a}$: $f(n) = O(n^3)$ and $n^{\log_b a} = n^2$

4. In this case $f(n) = O(n^3) = O(n^{2+e})$ and for $e =1$, this case falls under case 3 of the master theorem

5. Solution is $T(n) = \boxed{\Theta}(f(n)) = \boxed{\Theta}(n^3)$

So, $T(n) = O(n^3)$

Question 3

20 / 20 pts

For each function below, write its corresponding O() expression. **No justification is needed.**

a) $12 n \sqrt{n} + 2 n^2 + 1$

b) $n^{1000} + 2^n + 24$

c) $2n^4 + 250 n^2 \log n$

d) $n! + n^n$

Your Answer:

a) $O(n^2)$

b) $O(2^n)$

c) $O(n^4)$

d) $O(n^n)$

Question 4

10 / 10 pts

Let $f(n)= 3n^4+ n^4 \log n$. Then $f(n)= O$ (?), $f(n)= \Omega$ (?), $f(n)= \theta$ (?)

Replace question marks with the correct answer (we look for the **tightest** bound once possible). Choose the right case from the following options:

- $f(n)= O(n^4 \log n)$, $f(n)= \Omega(n^4 \log n)$, θ is not applicable
- $f(n)= O(n^4)$, $f(n)= \Omega(n^4)$, $f(n)= \theta(n^4)$
- $f(n)= O(\log n)$, $f(n)= \Omega(\log n)$, $f(n)= \theta(\log n)$
- None of the cases are correct.

- f(n)= O(n^4), f(n)= Ω(n^2), θ is not applicable
- f(n)= O(n^4), f(n)= Ω(n^4), f(n)= θ((n^4 log n))
- f(n)= O(n^4 log n), f(n)= Ω(n^4 log n), θ (n^4 log n)

Question 5

10 / 10 pts

How many times the operation inside the while loop will be executed? Assume n>=1 .

Always look for the tightest bound.

```
void fun (int n)
{
    int j = n;
    while (j >0)
        j = j/2;
}
```

- O(1)
- None of the answers are correct.
- O(j)
- O(n/2)
- O(n log n)
- O(n * j)
- O(n*2)
- O(n)
- O(log n)

Question 6

10 / 10 pts

Consider asymptotic growth of the following functions in the worst case.

Select the correct order among the following cases, where f1(n)<= f2(n) means f1(n) grows not faster than f2(n) in the worst case as n grows.

- ○ $O(n) <= O(\log n) <= O(n^4) <= O(n^n)$
- ○ None of the cases is correct.
- ○ $O(2^n) <= O(n^4) <= O(n!) <= O(n^n)$
- ○ $O(1) <= O(n) <= O(n^3) <= O(n \log n) <= O(n!)$
- ◉ $O(\log n) <= O(n) <= O(n \log n) <= O(n^{5/2}) <= O(2^n)$

Quiz Score: 80 out of 80