

# CSCI 271 Algorithm Analysis & Data Structures

## Spring 2026 — Assignment III

**Due by: Wednesday Feb. 25, 2026 11:59 PM**  
**PLEASE DO NOT include your WID in any submission**

### Question 1 [10 marks]

Write a **JAVA** program that reads in a string **S** from the user and displays the length of **S**. In doing so, the program will use a recursive function that takes **S** as an argument and returns the number of characters in **S** recursively! Solutions to this question using a loop are not acceptable! In addition, calculate the running time of this function and show your work in details.

### Question 2 [20 marks]

Write a **JAVA** program that reads a string **S** and a single character **C** from the user and displays the number of times **C** occurs in **S**. This program will use a function that takes **S** and **C** as arguments and returns the number of times **C** appears in **S** recursively! Solutions to this question which use a loop are not acceptable! In addition, calculate the running time of this function and show your work in details.

### Question 3 [10 marks]

Write a **JAVA** program that reads a list of integers from the user and stores them into an array **A**, then, the program will find the maximum integer in **A** using a recursive function **max()**. Solutions to this question using a loop are not acceptable!! In addition, calculate the running time of this function. Show your work in details.

### Question 4 [20 marks]

Write a **JAVA** program that reads in an integer **N** and a single digit **D** from the user and displays the number of times **D** occurs in **N**. The program will use a recursive function that takes **N** and **D** as arguments and returns the number of times **D** appears in **N** recursively! here are few examples:

**N = 312458713476217354872163548721**  
**D = 8**  
**Answer = 3**

**N = 312458713476217354872163548721**  
**D = 9**  
**Answer = 0**

Please note: your solution must be recursive! Solutions to this question using a loop are not acceptable! In addition, calculate the running time of this function. Show your work in details.

### Question 5 [20 marks]

Write a **JAVA** program that reads in a string **S** from the user and displays it backwards. The program must use a recursive function that takes **S** as an argument and returns **S'** as **S** but backwards. Solutions to this question using a loop are not acceptable! In addition, calculate the running time of this function and show your work in details.

### Question 6 [20 marks]

Write a **JAVA** program that reads a list of integers from the keyboard and stores them into an array **A**, then, the program will return the sum of all even integers in **A** using a recursive function **sumEven()**. The program will then print the resulting sum to the screen. In doing so, the function must be recursive and should not print anything to the screen, the main program will! Solutions to this question using a loop are not acceptable! In addition, calculate the running time of this function and show your work in details.

### What to Submit:

1. Upload your **source code** to your 271 assignment repository on GitHub and submit a link to it on blackboard.
2. Capture **the output of your test programs** for each question as text files (using the "script" command you learned in csci210) to show that your programs work correctly! **Submit this output file on blackboard. Screen shots are not acceptable!**
3. You may write the runtime analysis work as comments in your source code but you must indicate this clearly and in an organized manner in the solutions of each questions.

### Evaluation:

- Marks are allocated for each of the methods required by each question and for calculating the correct complexity of the algorithm it uses — calculate  $O()$  complexity for 10% of the marks.
- For each method, the **correctness** of the code will gain 70% of the marks allocated to it.
- Clarity of **comments and description** of methods and code segments will be graded at 10%
- **Test cases in the main program** will be graded with another 10% — please document these clearly. For each test case, please explain the condition it tests and how it does so briefly.