



UGANDA CHRISTIAN UNIVERSITY

A Centre of Excellence in the Heart of Africa

FACULTY OF ENGINEERING, DESIGN AND TECHNOLOGY

DEPARTMENT OF COMPUTING AND TECHNOLOGY

ADVENT 2023 SEMESTER EXAMINATION

PROGRAM: BSCS, BSDS

YEAR: 1 SEMESTER: 2

COURSE CODE: CSC1203

COURSE NAME: DATA STRUCTURES AND ALGORITHMS

EXAMINATION TYPE: THEORY EXAM

EXAMINATION DATE: DECEMBER 2023

TIME ALLOWED: 2 HOURS

Examination Instructions

1. The general Uganda Christian University examination guidelines and academic & financial policies apply to this examination. Violating any of the policies by the student automatically makes this examination attempt void, even if you have completed and submitted the answer booklet.
 2. This exam consists of **FOUR (4)** questions, and you are required to attempt only **THREE (3)** questions.
 3. Answers to every question should have a separate Python Code File and pseudocodes and algorithm analysis should be within the same file as comments.
 4. Uganda Christian University or its officials reserve the right of admission or disqualification from this exam.

Answer any THREE (3) questions.

Question 1 (20 Marks)

- a) Given the postfix expression (, is a separator):

[8 Marks]

4, 10, 5, +, *, 15, 3, /, -

Use a stack to evaluate the given postfix expression. Describe each step of your process.

- b) What is the equivalent infix expression?

[4 Marks]

- c) Convert the following infix to prefix and postfix:

[8 Marks]

A+B×(C-D)/E+F-G×H

Question 2 (20 Marks)

- a) State and explain 3 examples of linear data structures and 3 examples of non-linear data structures.

[12 Marks]

- b) In time complexity, why would you care about the growth rate as the input size increases instead of how long a piece of code takes to determine its time complexity? [4 Marks]

- c) What is the time complexity (complexity class) of the following segments of code (show how you reach your conclusion):

[4 Marks]

```
value = 0
i = n
while (i > 0):
    value += i
    i //= 2
```

Question 3 (20 Marks)

Consider the following mysterySort() function:

```
def mysterySort(lst):
    for i in range(len(lst)-1):
        k = i
        for j in range(i, len(lst)):
            if lst[j] < lst[k]:
                k = j
        lst[i], lst[k] = lst[k], lst[i]
```

- print(lst)
- a) Show the list contents, in order, displayed by all calls to print(lst) when mysterySort() is called with the input list lst = [3, 2, 8, 4, 6, 0, 7, 5, 1].
[10 Marks]
- b) What is the Big-O runtime complexity of mysterySort(), when called with an input list of length N?
[5 Marks]
- c) If you were given the choice to use this sort function or insertion sort, which one would you choose for optimal runtime performance across all types of input lists? Explain.
[5 Marks]

Question 4 (20 Marks)

- a) Write the pseudocode that performs a search for an array element based on its value or its index.
[5 Marks]
- b) Construct a Binary Search Tree by inserting the following sequence of numbers 10, 12, 5, 4, 20, 8, 7, 15 and 13.
[5 Marks]
- c) Write a Python code snippet of the class Node that implements a doubly linked list.
[5 Marks]
- d) Given the code snippet below, state its complexity class and why you say so.
[5 Marks]

```
lst= [14, 8, -23, 4, 6, 10, -18, 5, 5, 11]
maxSum= lst[0]
for i in range(0, len(lst)):
    sumz= 0
    for j in range(i, len(lst)):
        sumz+= lst[j]
        if sumz> maxSum:
            maxSum= sumz
print(maxSum)
```

-END OF EXAM QUESTIONS-
MERRY CHRISTMAS & HAPPY NEW YEAR!