

NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER II EXAMINATION 2023-2024

MH1403 – Algorithms and Computing

May 2024

TIME ALLOWED: 2 HOURS

INSTRUCTIONS TO CANDIDATES

1. This examination paper contains **SIX (6)** questions and comprises **FIVE (5)** printed pages.
2. Answer **ALL** questions. The marks for each question are indicated at the end of each question.
3. Answer each question beginning on a **FRESH** page of the answer book.
4. This is an **OPEN BOOK** exam. You can bring any printed and hand-written material to the examination hall.
5. Calculators are **NOT ALLOWED** in the exam.

Question 1. Python Class

- (i) What is the output of the following code? (5 marks)

```

class BAR:                                #line 1
    def __init__(self, data):             #line 2
        self.data = data                 #line 3
                                         #line 4
def f(w):                                  #line 5
    w.data = 3*w.data                     #line 6
    x = BAR(w.data)                       #line 7
    x.data = x.data-1                     #line 8
    return x                              #line 9
                                         #line 10
x = BAR(3)                                #line 11
f(x)                                       #line 12
print(x.data)                             #line 13

```

- (ii) What is the output of the following code? (5 marks)

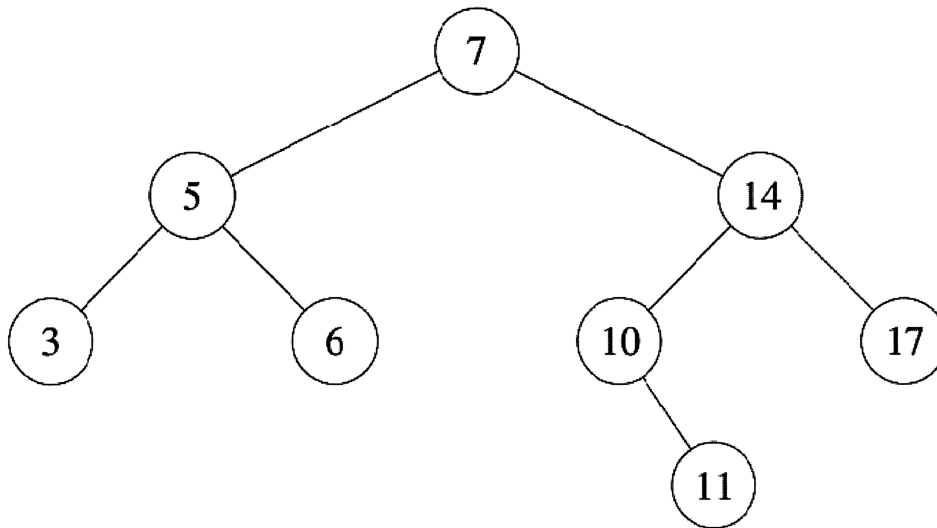
```

class BAR:                                #line 1
    def __init__(self, data):             #line 2
        self.data = data                 #line 3
    def g(self):                           #line 4
        self.data = self.data+1          #line 5
    def h(self):                           #line 6
        self.data = self.data*2          #line 7
                                         #line 8
class BAZ(BAR):                           #line 9
    def h(self):                           #line 10
        self.data = self.data*3          #line 11
                                         #line 12
x = BAZ(5)                                #line 13
x.g()                                     #line 14
x.h()                                     #line 15
print(x.data)                             #line 16

```

Question 2. AVL Tree

We are given the following AVL tree:



- (i) What is the preorder traversal of this AVL tree? (5 marks)
- (ii) Insert node 12 into this AVL tree. Draw the resulting AVL tree. (10 marks)
- (iii) Next, delete node 3 from the AVL tree obtained from Q2(ii). Draw the resulting AVL tree. (5 marks)

Question 3. For the list $\text{listA} = [3, 12, 6, 8, 10, 5, 2, 9, 7]$, use the last element as the pivot to partition the list. Rearrange the list so that all elements less than the pivot come before all elements greater than the pivot. What is listA after partitioning? You need to use the partitioning method taught in this course. You should give the details of the partitioning. (10 marks)

Question 4. Big-Oh Notation

- (i) What is the Big-Oh notation of the running time of the following function $f(n)$? (5 marks)

```
def f(n):
    s = 0
    for i in range(n):
        for j in range(100):
            s = s + (i*j)**0.5
    return s
```

- (ii) What is the Big-Oh notation of the running time of the following function $f(x, y, n)$ in terms of n ? Please explain how you obtained your result. (10 marks)

```
# x, y, n are positive integers
# x is an integer between 2 and 999,
# y is an integer between 1001 and 2000.
def f(x, y, n):
    if (n == 0):
        return 1
    elif (n%2 == 0):
        t = f(x, y, n/2)%y
        return t*t%y
    else:
        t = f(x, y, n//2)%y
        return x*t*t%y
```

- (iii) Given the recurrence relation $T(n) = 3T(n/4) + n$, compute the Big-Oh notation of $T(n)$. (5 marks)

Question 5. Given a sorted list `listA` containing n positive integers in ascending order, you need to write an **efficient** function to check if there exist two different indices i and j , where $i < j$, such that $\text{listA}[j] = 2 * \text{listA}[i]$. If such two indices exist, the function returns `True`; otherwise, it returns `False`. You need to provide the Big-Oh notation for the running time of your function. (20 marks)

Question 6. A bitonic sequence is a sequence of numbers that first increases and then decreases. Given a Python list listA containing n integers, you are required to develop and describe an **efficient** dynamic programming algorithm to find the length of the longest bitonic subsequence in listA (you do not need to find the longest bitonic subsequence itself). You need to give the Big-Oh notation of the running time of your algorithm. For example, the length of the longest bitonic subsequence of the sequence [4, 2, 5, 9, 7, 6, 10, 3, 1, 8] is 7 (which is the sequence [4, 5, 9, 7, 6, 3, 1]). Hint: We have learned how to find the length of the longest increasing subsequence. (20 marks)

END OF PAPER

MH1403 ALGORITHMS & COMPUTING

Please read the following instructions carefully:

- 1. Please do not turn over the question paper until you are told to do so. Disciplinary action may be taken against you if you do so.**
2. You are not allowed to leave the examination hall unless accompanied by an invigilator. You may raise your hand if you need to communicate with the invigilator.
3. Please write your Matriculation Number on the front of the answer book.
4. Please indicate clearly in the answer book (at the appropriate place) if you are continuing the answer to a question elsewhere in the book.