NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER II EXAMINATION 2022-2023

MH1403 – Algorithms and Computing

May 2023 TIME ALLOWED: 2 HOURS

INSTRUCTIONS TO CANDIDATES

1. This examination paper contains **FIVE (5)** questions and comprises **FOUR (4)** 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.

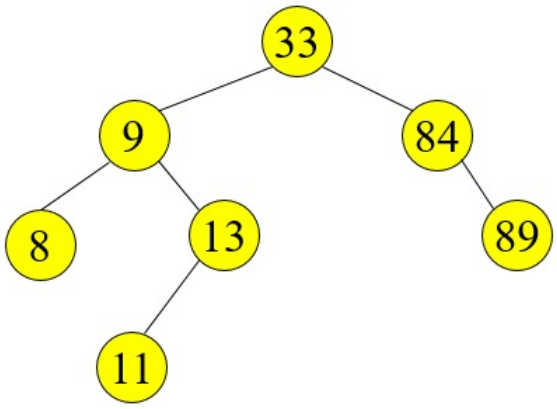
Question 1. In the following Python code, arr is a Python list.

(i) Let the input size m = right-left+1, what is the Big-Oh notation of the worst-case running time of the function maxCrossingSum(arr, left, mid, right)? (7 marks)

(ii) Let n denote the length of arr, what is the Big-Oh notation of the worst-case running time of the function maxSubSum(arr)? Please explain how you obtained the answer. (13 marks)

```
def maxCrossingSum(arr, left, mid, right):
    sum = 0
    #initialize left_sum as an extremely small value
    left_sum = float('-inf')
    for i in range(mid, left-1, -1):
        sum = sum + arr[i]
        if (sum > left_sum) :
            left_sum = sum

    sum = 0
    #initialize right_sum as an extremely small value
    right_sum = float('-inf')
    for i in range(mid + 1, right + 1) :
        sum = sum + arr[i]
        if (sum > right_sum):
            right_sum = sum

    return left_sum + right_sum

def _maxSubSum(arr, left, right):
    if (left == right) :
        return arr[left]

    mid = (left + right) // 2
    left_max = _maxSubSum(arr, left, mid)
    right_max = _maxSubSum(arr, mid+1, right)
    cross_max = maxCrossingSum(arr, left, mid, right)
    return max(left_max, right_max, cross_max)

# maxSubSum() and _maxSubSum() are two different functions
def maxSubSum(arr):
    return _maxSubSum(arr, 0, len(arr)-1)
```

Question 2. AVL Tree
We are given the following AVL tree:



(i) What is the postorder traversal of this AVL tree? (5 marks)

(ii) Delete the node 89 from this AVL tree. Draw the resulting AVL tree. (15 marks)

Question 3. Given a sorted list A of n distinct integers, arranged in increasing order, you want to find out whether there is an index i for which A[i] = i. Design and describe an efficient algorithm for this problem. You need to give the Big-Oh notation of the worst-case running time of your algorithm.

For example, for the Python list A = [-2, -1, 0, 1, 4, 7, 8, 10], it is true that we can find A[i] = i since A[4] = 4. (20 marks)

Question 4. Let A be a Python list of n distinct integers. A pair (A[i], A[j]) is said to be an inversion if i < j but A[i] > A[j]. Design and describe an efficient algorithm for counting the number of inversions of A. The Big-Oh notation of the worst-case running time of your algorithm needs to be O(n log(n)). (20 marks)

Question 5. Given a Python list A of n positive integers, and given a positive integer x, you are required to develop an **efficient** algorithm to determine if we can find a subset R of the elements of list A so that the sum of the elements in R is equal to x. Describe your algorithm and give the worst-case Big-Oh notation of your algorithm.

For example, A = [7, 3, 2, 1, 5, 4, 8], if x = 11, we can find the subset with the sum 11, R = {7, 3, 1}.

In your algorithm, you only need to determine whether such a subset R exists or not, and you do not need to find the elements of R. (20 marks)

END OF PAPER