CSE 321 – Introduction to Algorithm Design

Homework 02

Deadline: 4 November 23:55

PLEASE DO NOT FORGET TO READ THE NOTES CAREFULLY

THE NOTES ARE UPDATED

1. Watch the film "The Imitation Game" which is based on the biography of Alan Turing. Write your thoughts about the film using an office program. It will at least occupy half of the A4 paper. **(10 points)**
2. Solve following recurrence relations using master theorem. If any of the problems cannot be solved using master theorem, state that it can't be solved and explain the reason. **(18 points)**

$$x_1(n) = 0.5x_1\left(\frac{n}{2}\right) + \frac{1}{n}$$

$$x_2(n) = 3x_2\left(\frac{n}{4}\right) + nlogn$$

$$x_3(n) = 3x_3\left(\frac{n}{3}\right) + \frac{n}{2}$$

$$x_4(n) = 6x_4\left(\frac{n}{3}\right) + n^2 logn$$

$$x_5(n) = 4x_5\left(\frac{n}{2}\right) + \frac{n}{logn}$$

$$x_6(n) = 2^n x_6\left(\frac{n}{2}\right) + n^n$$

3. Consider the following algorithm implemented in python: **(18 points)**

```python
def chocolateAlgorithm(n):
    #Input is a positive integer n
    if n==1:
        return 1
    else:
        return chocolateAlgorithm(n-1) + 2 * n -1
```

a-) Set up a recurrence relation for this function's values and solve it to determine what this algorithm computes.

b-) Set up a recurrence relation for the number of multiplications made by this algorithm and solve it.

c-) Set up a recurrence relation for the number of additions/subtractions made by this algorithm and solve it

4. **(18 points)** Consider the problem of finding required time to finish Tower of Hanoi problem with optimal number of moves recursively. The time takes to move a disk is calculated as the weight of a disk (n) multiplied by the distance between the source and destination pegs (can be 1 or 2). For example, to move disk 2 from peg 1 to peg 3 takes 2*2=4 seconds. Design and implement a Python 3 function that simulates the gameplay by printing each move to console, and at last, prints the total elapsed time for moving each disk separately. Write the function in a python file named TOHtime[StudentID].py. Example output:

```
Input size is 3

disk 1: SRC to DST

disk 2: SRC to AUX

disk 1: DST to AUX

disk 3: SRC to DST

disk 1: AUX to SRC

disk 2: AUX to DST

disk 1: SRC to DST

Elapsed time for disk 1: 6

Elapsed time for disk 2: 4

Elapsed time for disk 3: 6
```

5. **(18 points)** Consider the problem of finding rotten walnut. You have n walnuts and the weights of all walnuts are equal except the rotten one which is lighter than the others. Your input will be a python list of positive integers which indicates the weight of all walnuts; for example: [1 1 1 1 1 0.5 1 1 1]. Your output (return value of function) will be the index of the rotten walnut. To find the rotten walnut, you will use a pair of scales. The Python function which compares two set of walnuts is:

```
def compareScales (leftScaleList, rightScaleList):
    result = sum(leftScaleList) - sum(rightScaleList)
    if result < 0:
        return 1
    elif result > 0:
        return -1
    else:
        return 0
```

You will assume that this function executes in constant time (O (1)).

a) Using this function, design a recursive algorithm that finds the index of rotten walnut **faster than linear time(O(n)) in terms of asymptotic complexity**. Write a python function of the algorithm you designed in a python file named findRottenWalnut[StudentID].py. Your function will be tested with different inputs.

b) Explain your algorithm. Show this algorithm's number of operations in terms of input array size (n) and complexity using the asymptotic notations for best and worst cases.

6. **(18 points)** Solve the following recurrence relations,

a) Using forward/backward substitution:

$T_1(n) = 3T_1(n-1) \qquad for\ n > 1,\ T_1(1) = 4$

$T_2(n) = T_2(n-1) + n \qquad for\ n > 1,\ T_2(0) = 0$

$T_3(n) = T_3(n/2) + n \qquad for\ n > 1,\ T_3(1) = 0\ (\,solve\ for\ n = 2^k)$

b) Using the properties of linear homogeneous/inhomogeneous equations:

$T_1(n) = 6T_1(n-1) - 9T_1(n-2)\ ,\ T_1(0) = 1\,, T_1(1) = 6$

$T_2(n) = 5T_2(n-1) - 6T_2(n-2) + 7^n\ ,$

$(Find\ the\ particular\ solution\ and\ then\ \boldsymbol{form}\ of\ the\ real\ solution)$

IMPORTANT NOTES

1. You will write your answers in Microsoft Word or similar office program. No photos, no handwritten papers.
2. Homeworks will be submitted in ZIP format. The file hierarchy will be this:
   CSE321_HW2_[StudentID].zip
   | →CSE321_HW2_ANS_[StudentID].pdf
   | → iterativeTOH[StudentID].py
   | → findRottenWalnut[StudentID].py
3. Use homework question forum on moodle if you have any questions about homework.
4. Cheating will be punished. (Grade will be -100)
5. No late submissions.