

CSE 321 – Fall 2023

Homework 2

Due Date: (19/11/2023) 23:55

1. Solve the following recurrence relations and provide a Θ bound for each of them. You must use backward substitution, forward substitution or the Master's Theorem at least once to solve the following relations.

- a) $T(n) = 3 * T(n - 1) - 2 * T(n - 2)$
- b) $T(n) = T(n/2) + 1$
- c) $T(n) = 4T(n - 1) - 4T(n - 2) + 3n$
- d) $T(n) = 4T(n/2) + n^2$
- e) $T(n) = 2T(n/2) + O(n)$
- f) $T(n) = T(n/2) + T(n/4) + n$
- g) $T(n) = T(n/2) + n$
- h) $T(n) = 2T(\sqrt{n}) + 1$

2. Provide pseudo code for the following operations on a given binary search tree (BST) with n nodes. Derive a recurrence relation for each of your algorithms. Calculate the average-case $\Theta()$ complexity of the derived recurrence relations.

- a) `is_balanced(BST)`: This function checks whether the given binary search tree is balanced or not.
- b) `height_of_tree(BST)`: This function returns the height of the given binary search tree.

3. Suppose you are choosing between the following three algorithms:

- a) Algorithm A divides a problem with size n into five sub problems that are one-half the size, solves each one recursively, and then combines the results in cubic time.
- b) Algorithm B solves a problem with size n by resolving two sub problems of size $n-2$ recursively and then integrating the solutions in linear time.
- c) Algorithm C addresses issues of size n by dividing them into three subproblems, each half the size, solving each subproblem recursively, and then combining the solutions in $O(n^2)$ time.

What is the running time of each algorithm (in terms of big -Oh notation), and which one would you choose? Provide a detailed explanation for your choice.

4. The maximum cardinality matching problem in bipartite graphs involves finding the largest possible set of pairwise non-adjacent edges in a given bipartite graph. A bipartite graph is a graph in which the set of vertices can be divided into two disjoint sets, A and B, such that all edges connect a vertex from set A to a vertex in set B (and vice versa).

Provide a polynomial-time algorithm to compute a maximum cardinality matching in bipartite graphs and analyze the worst-case, best-case and average-case time complexity of the algorithm.

5. Write a recurrence relation to calculate the number of characters printed when the following function is called with input n .

```
foo(n):  
  
    if n <= 1:  
        return 1  
    else:  
        for i in range(n):  
            print("a")  
        return foo (n / 2) + foo(n/2)
```

Notes:

- Your answer must be handwritten and submitted via the Course MS Teams page.
- Pseudocodes should be submitted as actual Python code and submitted as separate files together with your handwritten solutions.
- If you have any questions, you can send an email to b.koca@gtu.edu.tr
- Please complete your homework individually; group studies will be regarded as cheating.