

Homework 3 (50 points) Due: September 27, 2024 11:59 pm
COMPSCI 733: Advanced Algorithms and Designs

Documentation: (5 points) Type your solutions using Latex (www.overleaf.com or https://www.latex-project.org/). Submit your solutions (pdf is enough) to Canvas.

Problem 1: (15 points)

- (a) Arrange the following functions into increasing order of time complexity; that is, $f(n)$ should come before $g(n)$ in your list if $f(n)$ is $O(g(n))$. (i.e, n comes before n^2 since n is $O(n^2)$.)

$$100000, \log(n), 2^n, n \log(n), n^{0.9999999} \log(n), n + \log(n), n^{50} - 100n^2, \\ n^n, n!, 1.0000001^n, (\log(n))^n$$

- (b) Use the master theorem (the version given on the slides, not the CLRS textbook version) to give a tight bound (i.e., $\Theta()$) for the following recurrence relations. Write your steps using the format.

$$T(n) = aT(n/b) + f(n), a \geq 1, b > 1.$$

- Identify $a, b, f(n)$.
- Calculate $n^{\log_b a}$
- Compare $n^{\log_b a}$ with $f(n)$ and identify a case from the theorem.
- For Case 3 only: Check the regularity condition. i.e., Find $c < 1$ such that $af(n/b) \leq cf(n)$.
- Conclusion: The Θ bound or "The theorem cannot be applied."

- (a) $T(n) = 2T(\frac{n}{4}) + 1$
- (b) $T(n) = 2T(\frac{n}{4}) + \sqrt{n}$
- (c) $T(n) = 2T(\frac{n}{2}) + n^2$
- (d) $T(n) = 4T(\frac{n}{2}) + n^2 \log n$

Problem 2: (15 points) An element x of an array $A[1 : n]$ is called a majority element if more than half of A are equal to x . Your task is to find whether a given array has a majority element and if so, what it is. We do not assume the elements of A are integers, or otherwise ordered; the only query you can access the elements of A is of the form whether $A[i] = A[j]$ in constant time, for any i and j .

- (a) Suppose that A has a majority element, x . Let L and R partition A ; i.e., L and R are disjoint subsets of A such that their union is A . Use proof by contradiction to prove that if x is a majority element in A , then it must also be a majority element in L or in R .

Proof. (By contradiction)

Suppose x is a majority element in A and x is not a majority element in both L and R .

***Complete the proof by finding a contradiction. ***

□

- (b) Solve this problem with a Divide-and-Conquer algorithm that runs in time $O(n \log n)$. Write a pseudo code of your algorithm. (Hint: By dividing A into two halves A_1 and A_2 , and using Part (a). You can use English statements to explain the steps of your pseudo code. Direct copying of online versions is not accepted.

Algorithm 1 Majority(A)

1: Your pseudocode goes here.

- (c) write a recurrence relation for the execution time, $T(n)$, and apply Master's theorem to prove Θ time complexity of $T(n)$.

Problem 3: (15 points)

The Towers of Hanoi puzzle has 3 towers labeled 0, 1, and 2. There are n disks of sizes 1 to n , initially placed on tower 0 from top to bottom in the order of their sizes. The goal is to move all n disks from tower 0 to tower 2 under certain rules. One can only move one disk at a time, and move the top disk on a tower to be placed on the top of another tower, and cannot place a larger disk on top of a smaller disk. The question is, what is the smallest number of moves needed. Our problem is a variant of the Towers of Hanoi puzzle: The three towers are still labeled 0, 1, and 2. There is one additional constraint: You are only allowed to move disks from a tower labeled i to the next one labeled $(i + 1) \bmod 3$. You can imagine the towers as being arranged in a circular fashion. So, for example, in order to move a disk from tower 0 to tower 2, you would have to first move it to tower 1 (if both $0 \rightarrow 1$ and $1 \rightarrow 2$ are legal moves at this point), and this sequence would cost two moves instead of just one.

- (a) Explain that starting at any legal configuration, there is at least one legal move. (Any legal configuration means several legal moves may have already been made. Any tower may be empty or non-empty.)
- (b) Inductively prove that for any $n \geq 1$, this variant of the Towers of Hanoi puzzle has a solution consisting of finitely many legal moves.

$P(n)$: This variant of the Towers of Hanoi puzzle with n disks has a solution consisting of finitely many legal moves.

Prove: $P(n)$ is true for all $n \geq 1$.

Proof. (By induction on n)

- Prove the base case.
- Write the inductive hypothesis ($n = k$).
- Prove the inductive step. (Hint: Start with $n = k + 1$ disks.)
- Conclusion. (Given) Therefore, by the principle of mathematical induction, $P(n)$ is true for all $n \geq 1$.

□

- (c) Follow the given steps to show a recurrence of $T(n)$, the number of moves needed for the variant of the Towers of Hanoi puzzle with n disks, satisfies:

$$T(0) = 0, T(1) = 2, T(n) = 2T(n-1) + 2T(n-2) + 3.$$

- (a) Introduce a quantity $S(n)$ for the number of moves needed to move n pegs from tower 0 to tower 1 (or i to $(i+1) \bmod 3$). First show that $S(n) = 2T(n-1) + 1$. Explain your moves.
- (b) Derive $T(n)$ using $T(n-1)$ and $S(n-1)$.
- (c) Use Part (a) to derive $T(n)$ using $T(n-1)$ and $T(n-2)$.