

Analysis of Algorithm

Lab 03: Analyzing the Towers of Hanoi – Recurrence Relations and Complexity

Andrés Oswaldo Calderón Romero, Ph.D.

August 20, 2025

1 Introduction

This lab focuses on solving Problem Set 1 from the Introduction to Algorithms (6.046J) course offered by MIT OpenCourseWare. The problem set consists of exercises from the CLRS textbook (Introduction to Algorithms, 3rd Edition) and additional problems related to Asymptotic Notation and Recurrences.

The primary objective of this lab is to strengthen your understanding of fundamental algorithmic analysis techniques. By working through these exercises, you will:

1. Practice applying asymptotic notation to describe algorithm efficiency.
2. Develop intuition for solving recurrence relations that arise in divide-and-conquer algorithms.
3. Gain experience with formal proofs and justifications related to algorithmic complexity.

We strongly encourage you to attempt the exercises independently before consulting external resources such as solutions or AI-based tools. The goal is not just to obtain correct answers but to understand the problem-solving process and improve your analytical thinking.

Problem Set 1 comes from the [Introduction to Algorithms 6.046J](#) course of the MIT OpenCourseWare Initiative. The problem set is available at this [link](#).

Here we go!

2 Exercises from the CLRS Textbook

We will use the **3rd edition** of the [Introduction to Algorithms](#) textbook. We will solve the set of exercises listed in Table 1.

Exercise	Page
2.3-3	39
2.3-4	39
2.3-5	39
2.3-6	39
3.1-1	52
3.1-2	52
3.1-3	53
3.1-4	53
3.1-5	53
3.1-6	53
3.2-2	60
3.2-3	60
3.2-4	60
4.1-3	74
4.1-4	74

Table 1: Exercises from CLRS textbook **3rd Edition**.

3 Asymptotic Notation

For each of the following statements, decide whether it is **always true**, **never true**, or **sometimes true** for asymptotically nonnegative functions f and g . If it is **always true** or **never true**, explain why. If it is **sometimes true**, give one example for which it is true, and one for which it is false.

- a) $f(n) = O(f(n)^2)$
- b) $f(n) + g(n) = \Theta(\max(f(n), g(n)))$
- c) $f(n) + O(f(n)) = \Theta(f(n))$
- d) $f(n) = \Omega(g(n))$ and $f(n) = o(g(n))$ (note the little- o)
- e) $f(n) \neq O(g(n))$ and $g(n) \neq O(f(n))$

4 Recurrences

Give asymptotic upper and lower bounds for $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for $n \leq 10$. Make your bounds as tight as possible, and justify your answers.

- a) $T(n) = 2T(\frac{n}{3}) + n \lg n$
- b) $T(n) = 3T(\frac{n}{5}) + \lg^2 n$
- c) $T(n) = T(\frac{n}{2}) + 2^n$

- d) $T(n) = T(\sqrt{n}) + \Theta(\lg \lg n)$
- e) $T(n) = 10T(\frac{n}{3}) + 17n^{1.2}$
- f) $T(n) = 7T(\frac{n}{2}) + n^3$
- g) $T(n) = T(\frac{n}{2} + \sqrt{n}) + \sqrt{6046}$
- h) $T(n) = T(n - 2) + \lg n$
- i) $T(n) = T(\frac{n}{5}) + T(\frac{4n}{5}) + \Theta(n)$
- j) $T(n) = \sqrt{(n)}T(\sqrt{(n)}) + 100n$

5 What we expect

You will compile all your answers into a well-structured **handwritten** report and submit it by **September 4, 2025**. Please scan your report in **PDF format** and send it via Brightspace.

As in previous labs, we already know that there are solutions for all the exercises proposed in the problem set. The MIT OpenCourseWare offers solutions for sections [3](#) and [4](#). Similarly, solutions for the exercises in section [2](#) can be found in the Instructor's Manual of the textbook and even online. The same applies to the use of AI frameworks.

We are not concerned about whether you have access to the solutions, but we strongly encourage you to attempt the exercises on your own first before looking at the answers. There is no point in going directly to the answer or asking an AI for it if you do not try solving it yourself or at least understand the logical procedure behind the solution.

Happy Hacking! 😊