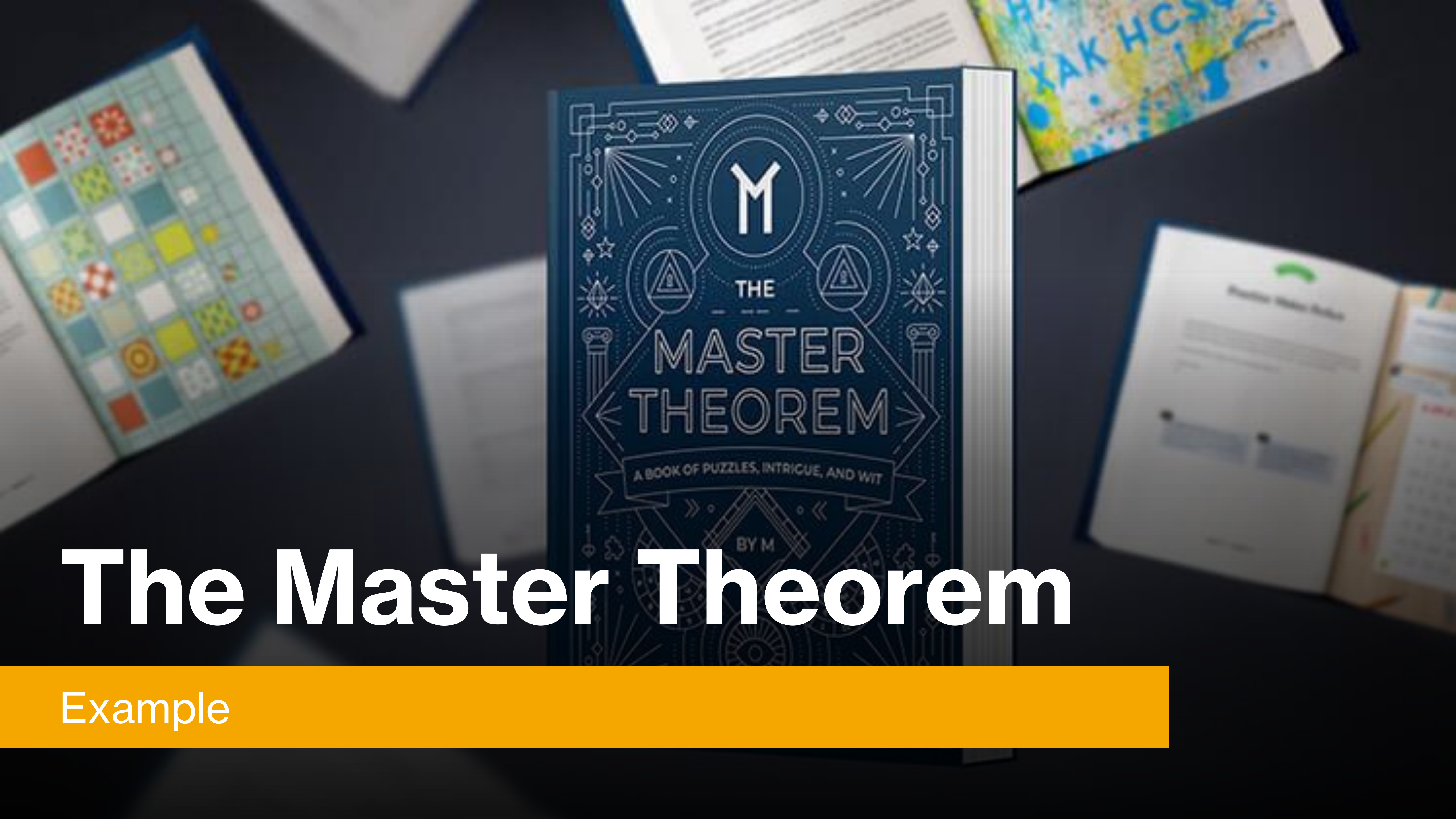

The Master Theorem

Example

Contents

- What is the Master Theorem
- Proof of Master Theorem

Writing a recurrence relation

$$T(n) = T\left(\frac{n}{2}\right) + O(1)$$

$$\downarrow$$

$$T(n) = O(\log n)$$

Binary Search Example

We have to create a recurrence tree to find the solution

$$T(n) = 4T\left(\frac{n}{2}\right) + O(n)$$

$$\downarrow$$

$$T(n) = O(n^2)$$

Polynomial Multiplication Example

We broke our problem into four sub-problems, each half the size, and did a linear amount of work.

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

$$\downarrow$$

$$T(n) = O(n^{\log_2 3})$$

When we had the more efficient algorithm, where we had only three sub-problems

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$

$$\downarrow$$

$$T(n) = O(n \log n)$$

Sometimes we break a problem into only two subproblems

Master Theorem

Theorem

If $T(n) = aT\left(\left\lceil \frac{n}{b} \right\rceil\right) + O(n^d)$ (for constants $a > 0, b > 1, d \geq 0$), then:

$$T(n) = \begin{cases} O(n^d) & \text{if } d > \log_b a \\ O(n^d \log n) & \text{if } d = \log_b a \\ O(n^{\log_b a}) & \text{if } d < \log_b a \end{cases}$$

Example 1

$$T(n) = 4T\left(\frac{n}{2}\right) + O(n)$$

$$a = 4$$

$$b = 2$$

$$d = 1$$

Since $d < \log_b a$, $T(n) = O(n^{\log_b a}) = O(n^2)$

Example 2

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

$$a = 3$$

$$b = 2$$

$$d = 1$$

Since $d < \log_b a$,

$$T(n) = O(n^{\log_b a}) = O(n^{\log_2 3})$$

Example 3

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$

$$a = 2$$

$$b = 2$$

$$d = 1$$

Since $d = \log_b a$,
$T(n) = O(n^d \log n) = O(n \log n)$

Example 4

University of Windsor

$$T(n) = T\left(\frac{n}{2}\right) + O(1)$$

$$a = 1$$

$$b = 2$$

$$d = 0$$

Since $d = \log_b a$, $T(n) = O(n^d \log n) = O(n^0 \log n) = O(\log n)$

Example 5

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

$$a = 2$$

$$b = 2$$

$$d = 2$$

Since $d > \log_b a$, $T(n) = O(n^d) = O(n^2)$