

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

# طراحی الگوریتم (قضیه اصلی)

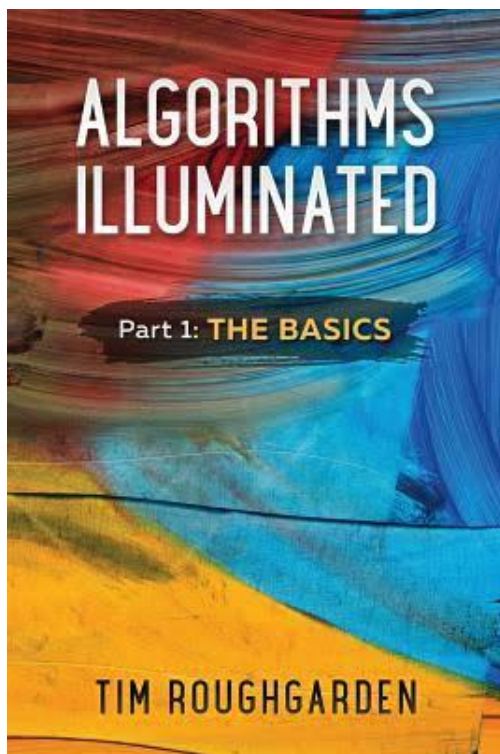


دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان

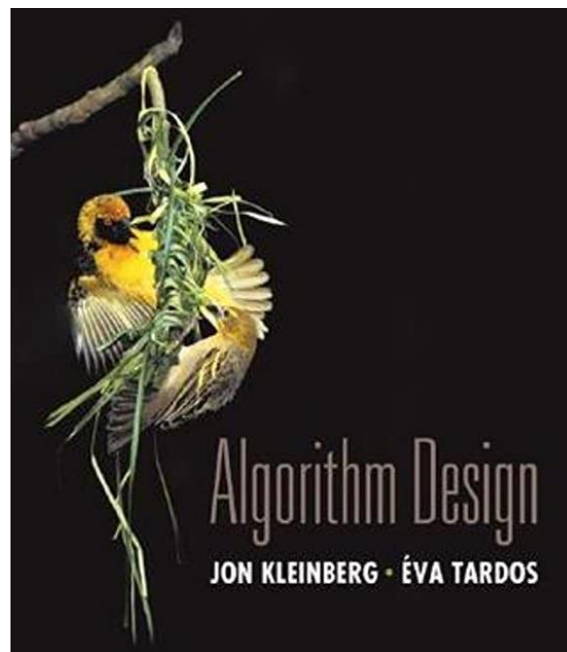
بهار ۱۴۰۰



# تقسیم و غلبه (یادآوری)



فصل چهارم، صفحه ۹۲



تقسیم مساله به زیرمساله‌های کوچکتر

غلبه و حل زیرمساله‌ها

ادغام جواب زیرمساله‌ها



## تحلیل زمانی ضرب اعداد



# فرم رابطه‌های بازگشتی متداول در تحلیل زمانی

## Standard Recurrence Format

**Base case:**  $T(n)$  is at most a constant for all sufficiently small  $n$ .<sup>4</sup>

**General case:** for larger values of  $n$ ,

$$T(n) \leq a \cdot T\left(\frac{n}{b}\right) + O(n^d).$$

**Parameters:**

- $a$  = number of recursive calls
- $b$  = input size shrinkage factor
- $d$  = exponent in running time of the “combine step”



# فرم رابطه‌های بازگشتی متداول در تحلیل زمانی



## قضیه اصلی



**Theorem 4.1 (Master Method)** *If  $T(n)$  is defined by a standard recurrence, with parameters  $a \geq 1$ ,  $b > 1$ , and  $d \geq 0$ , then*

$$T(n) = \begin{cases} O(n^d \log n) & \text{if } a = b^d \quad [\text{Case 1}] \\ O(n^d) & \text{if } a < b^d \quad [\text{Case 2}] \\ O(n^{\log_b a}) & \text{if } a > b^d \quad [\text{Case 3}]. \end{cases} \quad (4.2)$$

## کاربردهای قضیه اصلی



**Theorem 4.1 (Master Method)** *If  $T(n)$  is defined by a standard recurrence, with parameters  $a \geq 1$ ,  $b > 1$ , and  $d \geq 0$ , then*

$$T(n) = \begin{cases} O(n^d \log n) & \text{if } a = b^d \quad [\text{Case 1}] \\ O(n^d) & \text{if } a < b^d \quad [\text{Case 2}] \\ O(n^{\log_b a}) & \text{if } a > b^d \quad [\text{Case 3}]. \end{cases} \quad (4.2)$$

## اثبات قضیه اصلی





**Theorem 4.1 (Master Method)** *If  $T(n)$  is defined by a standard recurrence, with parameters  $a \geq 1$ ,  $b > 1$ , and  $d \geq 0$ , then*

$$T(n) = \begin{cases} O(n^d \log n) & \text{if } a = b^d \quad [\text{Case 1}] \\ O(n^d) & \text{if } a < b^d \quad [\text{Case 2}] \\ O(n^{\log_b a}) & \text{if } a > b^d \quad [\text{Case 3}]. \end{cases} \quad (4.2)$$

## اثبات قضیه اصلی



**Theorem 4.1 (Master Method)** *If  $T(n)$  is defined by a standard recurrence, with parameters  $a \geq 1$ ,  $b > 1$ , and  $d \geq 0$ , then*

$$T(n) = \begin{cases} O(n^d \log n) & \text{if } a = b^d \quad [\text{Case 1}] \\ O(n^d) & \text{if } a < b^d \quad [\text{Case 2}] \\ O(n^{\log_b a}) & \text{if } a > b^d \quad [\text{Case 3}]. \end{cases} \quad (4.2)$$

## اثبات قضیه اصلی