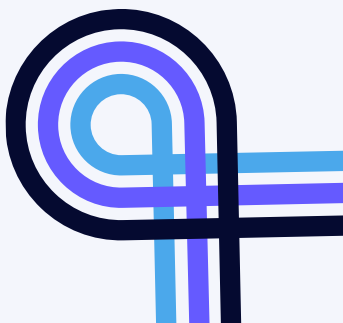


# Tutorial 4

## Analysis and Design of Algorithms

Divide and Conquer II



# Divide & Conquer Recap

- Divide & conquer is an **algorithmic paradigm** in which the problem is solved by **dividing** it into smaller parts, **conquering** them and possibly **combining** the sub problems together.
- In order to find an upper bound on the time needed to run a D&C algorithm, we write a **recurrence** and we solve it.

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ aT(b) + D(n) + C(n) & \text{otherwise} \end{cases}$$

where  $a$  is the number of subproblems,  $b$  is the size of each subproblem in terms of  $n$ ,  $D(n)$  is the divide time,  $C(n)$  is the combine time.

CSEN 703

L3- Divide and Conquer I

© Nourhan Ehab

**To solve the recurrence:**

1. Recursion Tree (last tutorial)
2. Master Theorem (this tutorial)

# Master Theorem

We use the master theorem to solve recurrences of the form  $T(n) = aT(n/b) + f(n)$ , where  $a \geq 1$  and  $b > 1$  and  $f(n)$  is asymptotically positive

## The Master Theorem

Let  $a \geq 1$  and  $b > 1$  be constants,  $f(n)$  be an asymptotically positive function, and  $T(n) = aT(\frac{n}{b}) + f(n)$ .

$T(n)$  can be asymptotically bounded in 3 cases:

- 1 If  $f(n) = O(n^{\log_b(a)-\varepsilon})$  for some  $\varepsilon > 0$ , then  $T(n) = \Theta(n^{\log_b(a)})$ .
- 2 If  $f(n) = \Theta(n^{\log_b(a)})$ , then  $T(n) = \Theta(n^{\log_b(a)} \log(n))$ .
- 3 If  $f(n) = \Omega(n^{\log_b(a)+\varepsilon})$  for some  $\varepsilon > 0$  and  $af(\frac{n}{b}) \leq cf(n)$  for  $0 < c < 1$ , then  $T(n) = \Theta(f(n))$ .

### Exercise 4-1

Use the master theorem to get the asymptotic notation of the following recurrences. If you fail to get the case, state the gap in which the recurrence falls.

i.  $T(n) = 2T(n/2) + n^3$

ii.  $T(n) = 16T(n/4) + n^2$

iii.  $T(n) = 8T(n/2) + n^2$

iv.  $T(n) = 5T(n/5) + \frac{n}{\lg(n)}$

v.  $T(n) = 5T(n/5) + n \lg(n)$



**Exercise 4-3** From CLRS (©MIT Press 2001)

The recurrence  $T(n) = 7T(n/2) + n^2$  describes the running time of an algorithm  $A$ . A competing algorithm  $A'$  has a running time of  $T'(n) = aT'(n/4) + n^2$ . What is the largest integer value for  $a$  such that  $A'$  is asymptotically faster than  $A$ .



**All done!**