CS 411 Analysis of Algorithms, Fall 2025

In-Class Worksheet 1: The Master Theorem—Solutions

Worksheet distributed in class on Monday, September 29.

1. Algorithm *A* is given a list as input. It uses a Divide and Conquer strategy. It splits its input in half (or nearly so), and handles each part with a recursive call. It also does other work requiring constant time.

Use the Master Theorem to determine the order of Algorithm *A*. As a guide through this process, answer each of the following.

Get b & a from the above description: 2 nearly equal-sized parts, 2 recursive calls.

b = 2; a = 2.

$f(n)$ is the amount of other work. This is constant time: $\Theta(1)$, that is, $\Theta(n^0)$.

$f(n)$ is $\Theta(n^d)$. d = 0.

$a = 2$. $b^d = 2^0 = 1$. So $a > b^d$.

Compare a and b^d. Circle one comparison operator: a < = [✓ >] b^d.

"<" gives case 1, "=" gives case 2, and ">" gives case 3.

Which case of the Master Theorem? Circle one: 1 2 [✓ 3]

Case 3 of the Master Theorem says $\Theta(n^k)$, where $k = \log_b a$. So $\Theta(n^1)$, that is, $\Theta(n)$: linear time.

Conclusion. The order of Algorithm *A* is (simplify!): $\Theta(n)$, or linear time.