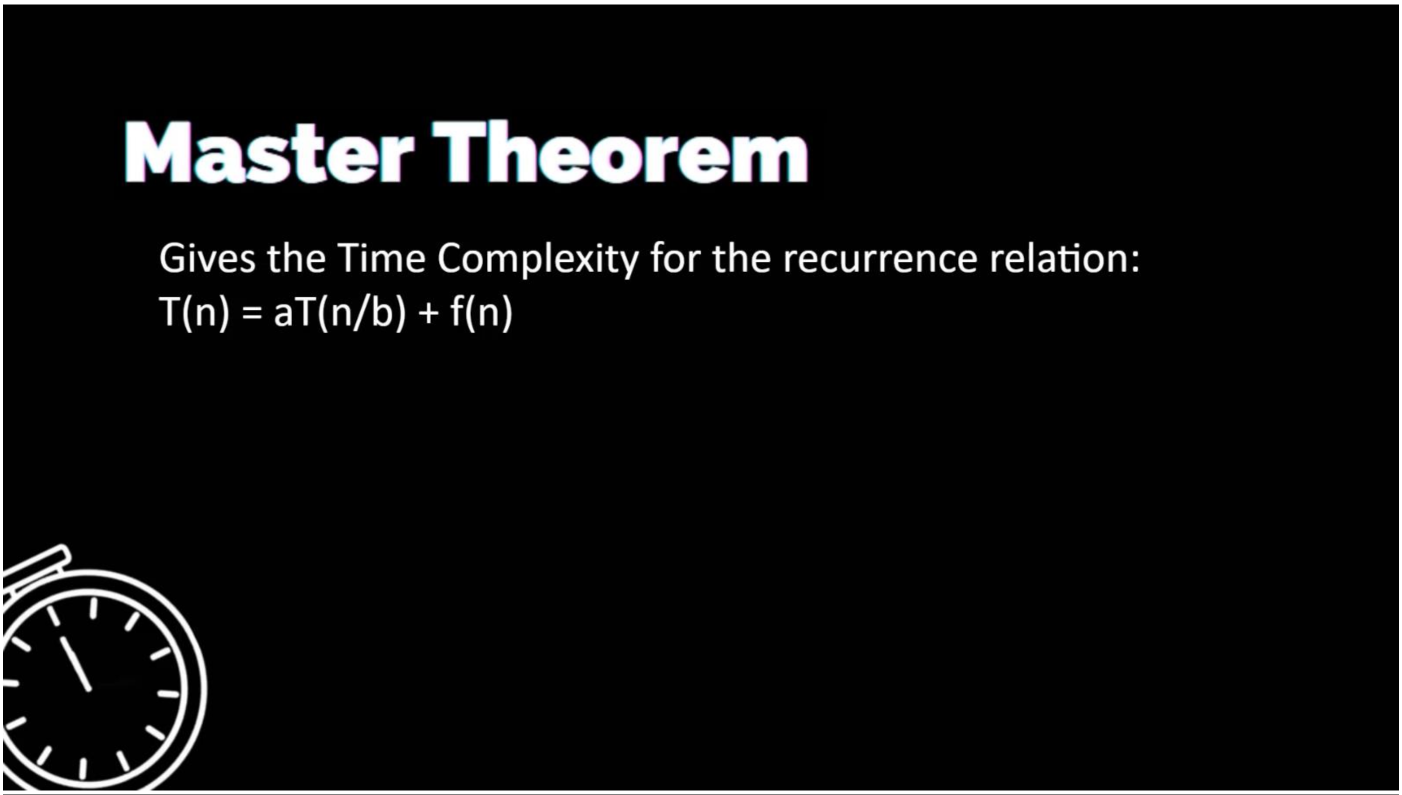
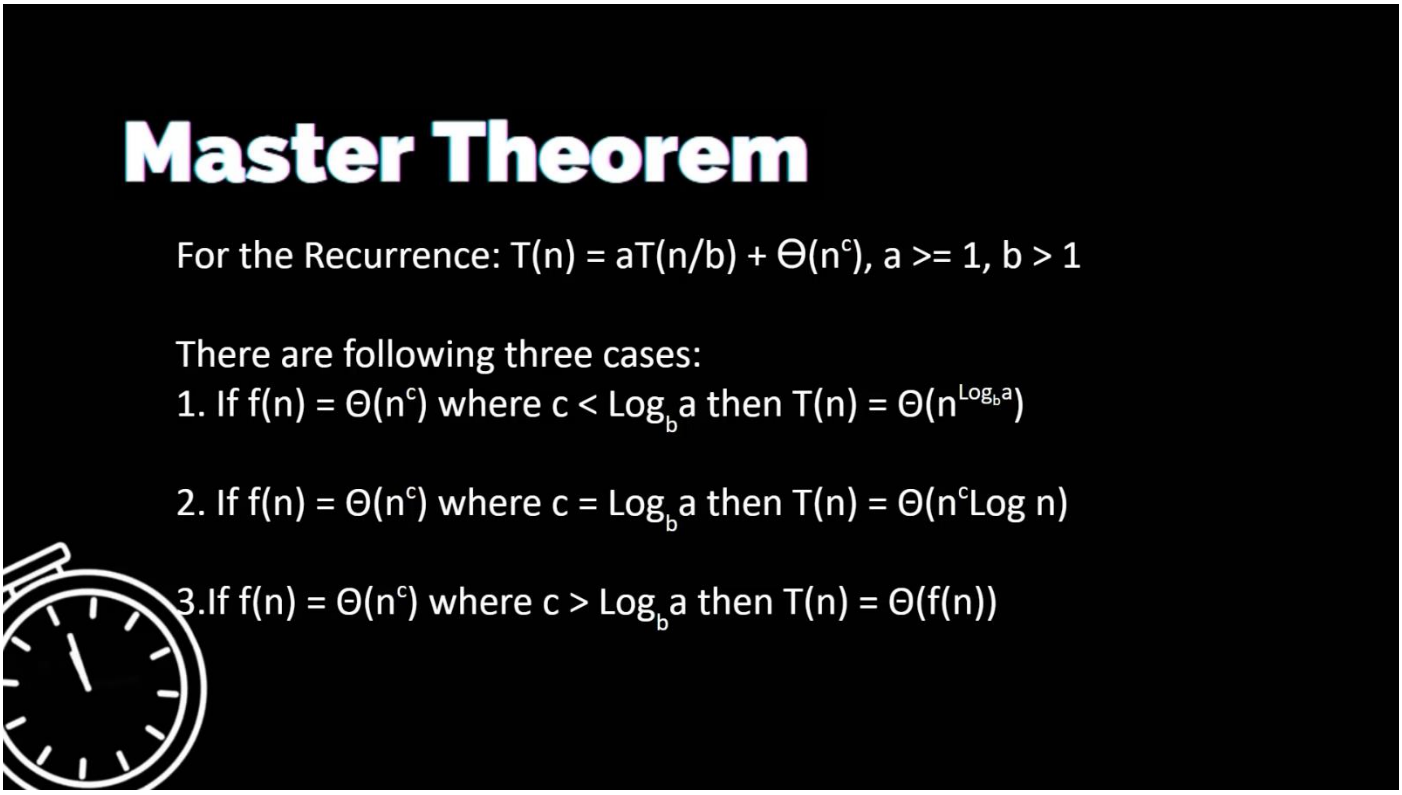
TIME COMPLEXITY OF A RECURRSIVE FUNCTION

MASTER THEOREM FOR





Problems:

1. $T(n) = 2\,T(n/2) + \Theta(n)$

$a = 2$, $b = 2$, $c = 1$
$\rightarrow c = \log_b a$

Time Complexity: $\Theta(n \,\text{Log}_2 n)$

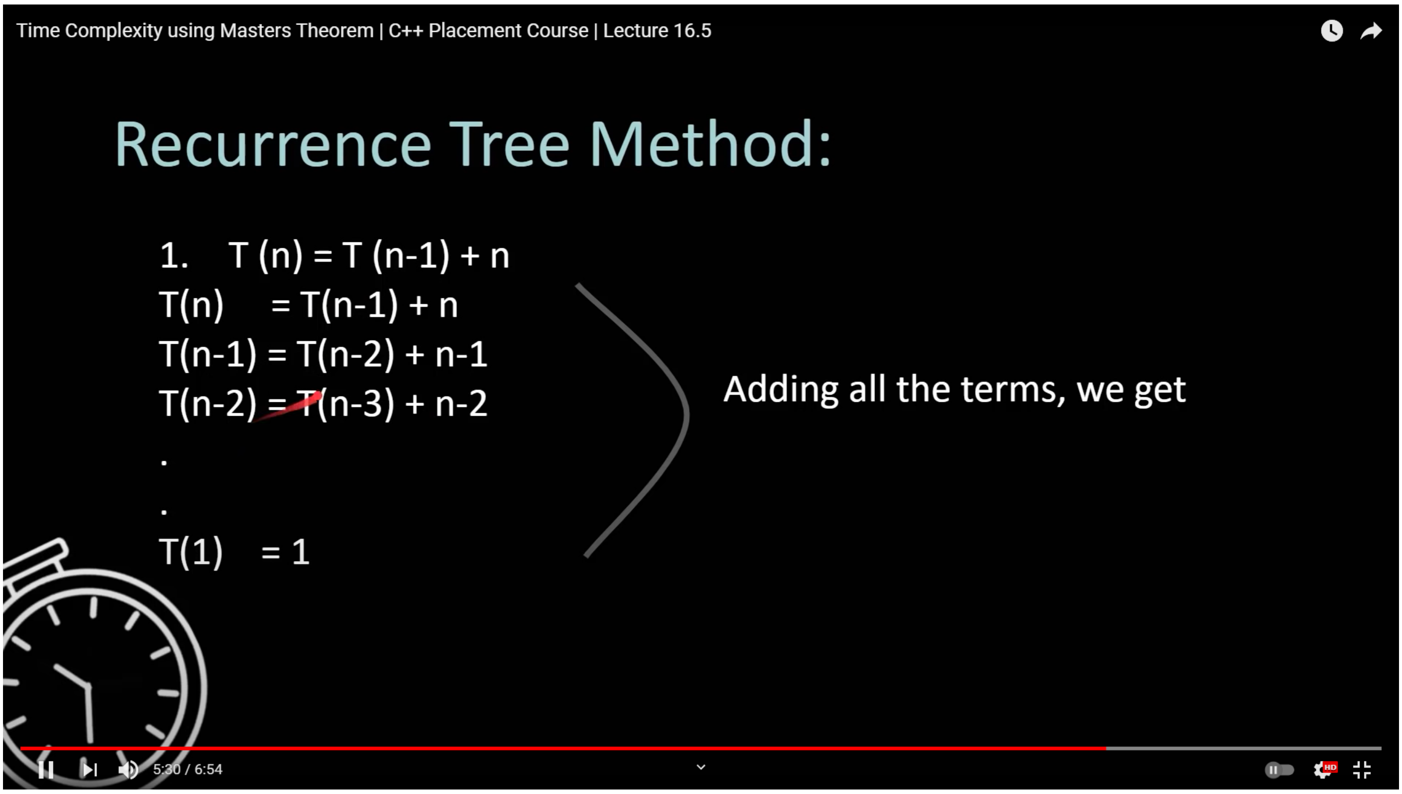
Problems:

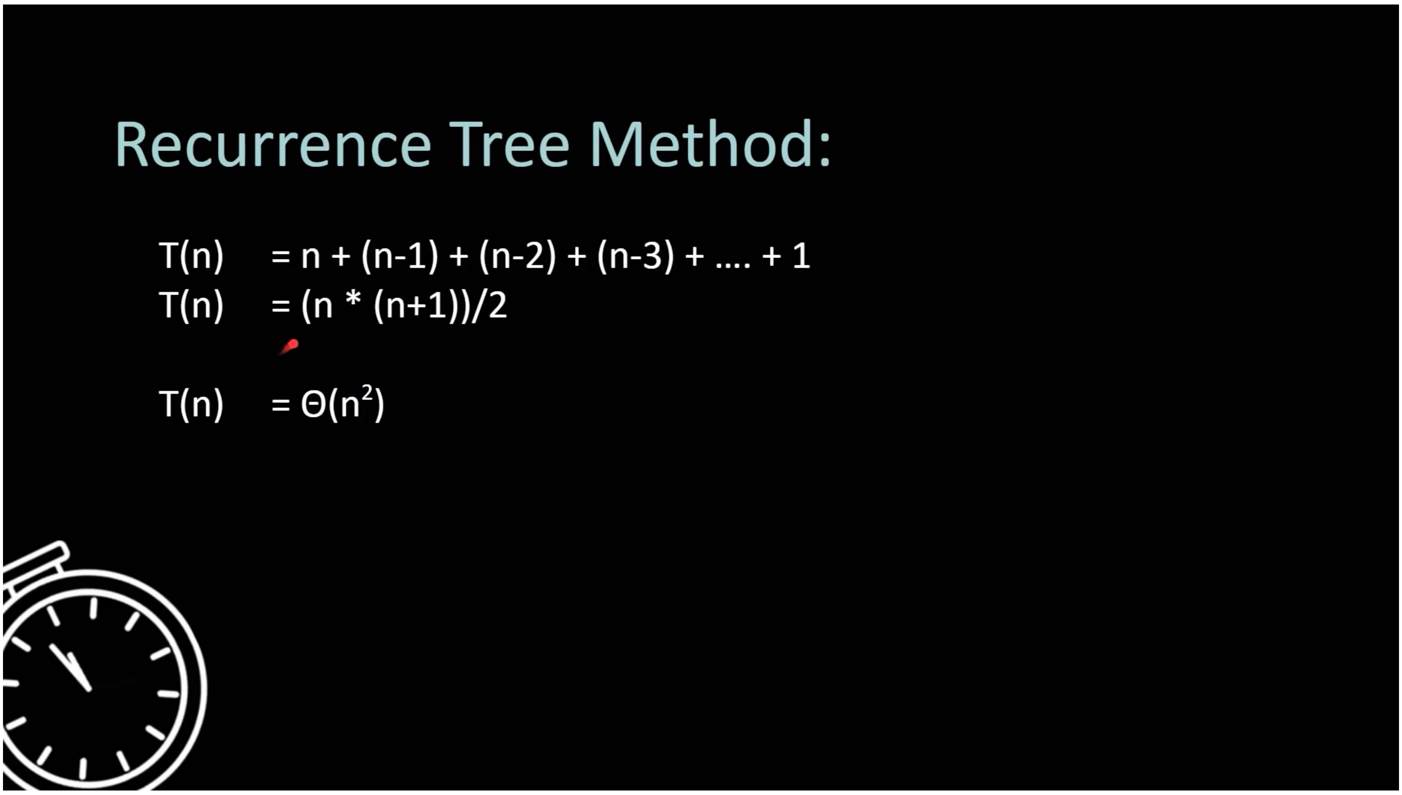
2. $T(n) = 3T(n/2) + n^2$

$a = 3$, $b = 2$, $c = 2$
$\rightarrow c > \log_b a$

Time Complexity: $\Theta(n^2)$

RECURRENCE TREE METHOD





Recurrence:

$T(n) = 2T(n/2) + n$

$T(n) = 2T(n/2) + n$

$T(n/2) = 2T(n/4) + n/2$

$T(n/4) = 2T(n/8) + n/4$

.

.

$T(1) = 1$

Quick Sort Complexity:

1. $T(n) = 2T(n/2) + n$

$T(n) = 2T(n/2) + n$ x1

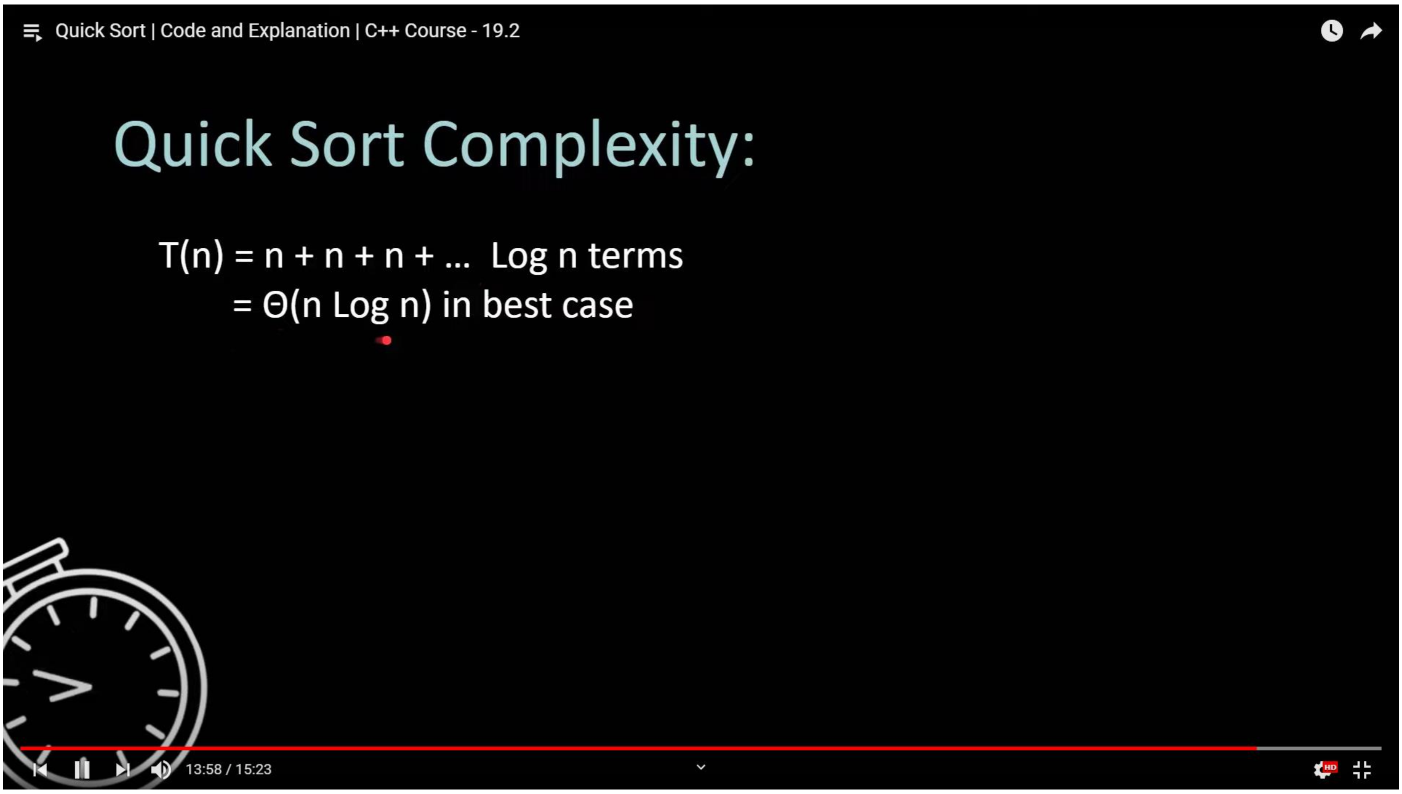
$T(n/2) = 2T(n/4) + n/2$ x2

$T(n/4) = 2T(n/8) + n/4$ x4

.   .

.   .

$T(1) = 1$ $x2^{\text{Log } n}$


Quick Sort | Code and Explanation | C++ Course - 19.2
Quick Sort Complexity:
T(n) = n + n + n + ... Log n terms
= Θ(n Log n) in best case
13:58 / 15:23