\[ 1 \ 2 \ 3 \ 4 \ \ldots \]

\[ t_1 \ t_2 \ t_3 \ t_4 \ \ldots \]

\[ \text{total time taken} = \sum_{i=0}^{n} t_i = X \]

\[ \text{total waiting time} = \# \text{ of people waiting for each time unit in } X \times \]

\[ = \sum_{i=0}^{n} \text{number of customers waiting} \]
how many people are waiting at time $t$

initially have all $n$ people waiting at any time we have $n$ - the number of people served
Binary search lower bound proof

not necessarily this specific algorithm

\[
A[\lceil n/3 \rceil] \leq 2
\]

yes / \ no

yes / \ no

each leaf is some index \( A[i] \) = \( \Theta(n) \) leaves

\( \ell (\log n) \) height = running time on some instance
Merging $k$ arrays of size $n$.

Note: This is the second half of merge sort.

height $= \log k n$

$\Theta(\log n)$

$\Theta(\log n)$

$\Theta(\log n)$

Total = $\Theta(\log n)$

Running time = $\Theta(n \log k n)$
\[ 3 \ 7 \ 5 \ 1 \ 6 \ 2 \ 3 \]
\[ \text{min} = 1 \ \text{max} = 7 \]

Sort: 1 2 3 4 5 6 7

3 1 10 106
\[ T(n) = 4 \cdot T\left(\frac{n}{3}\right) + n \log n \]

\[ a = 4 \quad \text{and} \quad b = 3 \quad \therefore f(n) = n \log n \]

\[ n^{\log_b a} = n^{\log_3 4} \quad \text{compare to} \quad f(n) \]

\[ f(n) = n \log n \quad n^{\log_3 4} \quad \log_3 4 > 1 \]

\[ n \log n = O\left(n \cdot n \text{ something}\right) \]

Case 1: \[ f(n) = O\left(n^{\log_b a - \varepsilon}\right) \quad \text{for} \quad \varepsilon > 0 \]

\[ \varepsilon = \log_3 4 - \log_3 3.5 \]

\[ f(n) = O\left(n^{\log_3 3.5}\right) \]

by the Master Theorem

\[ T(n) = \Theta\left(n^{\log_3 4}\right) = \Theta(n^{\log_3 4}) \]
\[ T(y) = 37 \left( \frac{y}{7} \right) + T \left( \frac{y}{5} \right) + \eta \]

This is not the right form.
\[ T(n) = 4T\left(\frac{n}{2}\right) + n^2 / \log n \]

\[ a = 4 \quad b = 2 \quad f(n) = n^2 / \log n \]

\[ n^{\log_2 4} = n^2 = n^2 \text{ compare to } n^2 / \log n \]

\[ n^2 / \log n = O(n^2) \]

Case 1: \( n \log n \)

\[ f(n) = O\left(n^{\log_2 n - \epsilon}\right) \]

Master Theorem does not apply because there is not a polynomial difference between \( f(n) \) and \( n^{\log_2 b} \).
\[ T(n) = q T\left(\frac{n}{3}\right) + n^a \]

\[ a = 9 \quad b = 3 \quad f(n) = n^2 \]

\[ n^\log_b q = n^{\log_3 9} = n^2 \]

\( f(n) \) compared to \( n^{\log_b q} \)

\[ n^2 = \Theta(n^2) \]

Case 2: \( T(n) = \Theta(f(n) \log n) = \Theta(n^a \log n) \)
\[ T(n) = 3T(n-12) + n^2 \]

\[ a = 3 \quad b = 12 \quad f(n) = n^2 \]

\[ n \not\in \Theta \]

The Master Theorem does not apply.
\[ T(n) = 3T\left( \frac{n}{4} \right) + n \]

\[ a = 3 \quad b = 4 \quad f(n) = n \]

\[ f(n) = n \quad n^\log_b a = n^\log_4 3 \]

\[ n = \Lambda\left( n^{\log_4 3} \right) < n \]

Case 3

\[ f(n) = \Lambda\left( n^{\log_b a + \varepsilon} \right) \quad n = \Lambda\left( n \right) \]

regularity condition \[ \varepsilon = 1 - \log_4 3 > 0 \]
regularity condition

\[ qf(n/b) \leq C \delta(h) \]

\[ 3(n/4) \leq c n \]

\[ \frac{3}{4} n \leq c n \]

\text{true}

Case 3: \[ T(n) = \Theta(f(n)) = n \]
\[ T(n) = 3T\left(\frac{n}{3}\right) + \Theta(n) \]

\[
\begin{array}{cccc}
\log n \\
\mid \downarrow \mid \\
\frac{n}{3} & \frac{n}{3} & \frac{n}{3} \\
\mid \downarrow \mid \downarrow \mid \\
\frac{n}{4} & \frac{n}{4} & \frac{n}{4} \\
\mid \downarrow \mid \downarrow \mid \downarrow \\
\frac{n}{4} & \frac{n}{4} & \frac{n}{4} \\
\end{array}
\]

The total work is\( \sum_{i=0}^{\log_2 n} a \frac{3^i}{2^i} \cdot n \)

Because this is increasing,

\[ \text{total work} = \sum_{i=0}^{\log_2 n} a \frac{3^i}{2^i} \cdot n \]

\[ = \Theta\left(\frac{3}{2} \cdot \log_2 n \right) \]

\[ = \Theta\left(\frac{3}{2} \cdot n \right) \]

\[ = \Theta\left(n \log_2 \left(\frac{3}{2} \cdot n \right) \right) \]
\[
\begin{align*}
\Theta(n - n \log_2 (3/2)) &= \Theta(n, n \log_2 3 - 1) \\
&= \Theta(n \log_2 3 - 1 + 1) \\
&= \Theta(n \log_2 3)
\end{align*}
\]