

Merge Sort is a D&C sorting algorithm much like Quicksort. Instead of Partition doing the sorting work as we go, however, we instead recursively reach the base cases and then merge subarrays together on the way back up.

MS(A, l, r)

If $l \geq r$

Return

MS(A, l, $\lfloor \frac{l+r}{2} \rfloor$)

MS(A, $\lfloor \frac{l+r}{2} \rfloor + 1, r$)

Merge(A, l, r)

Return

Merge(A, l, r)

Let B be an array of length $r-l+1$

Let $l_p = l, r_p = \lfloor \frac{l+r}{2} \rfloor + 1$, and $b_p = 0$

While $b_p < |B|$

If $l_p > \lfloor \frac{l+r}{2} \rfloor$

B[b_p] = A[r_p]

b_p ++

r_p ++

Else if $r_p > r$

B[b_p] = A[l_p]

b_p ++

l_p ++

Else if $A[l_p] \leq A[r_p]$

B[b_p] = A[l_p]

b_p ++

l_p ++

Else

B[b_p] = A[r_p]

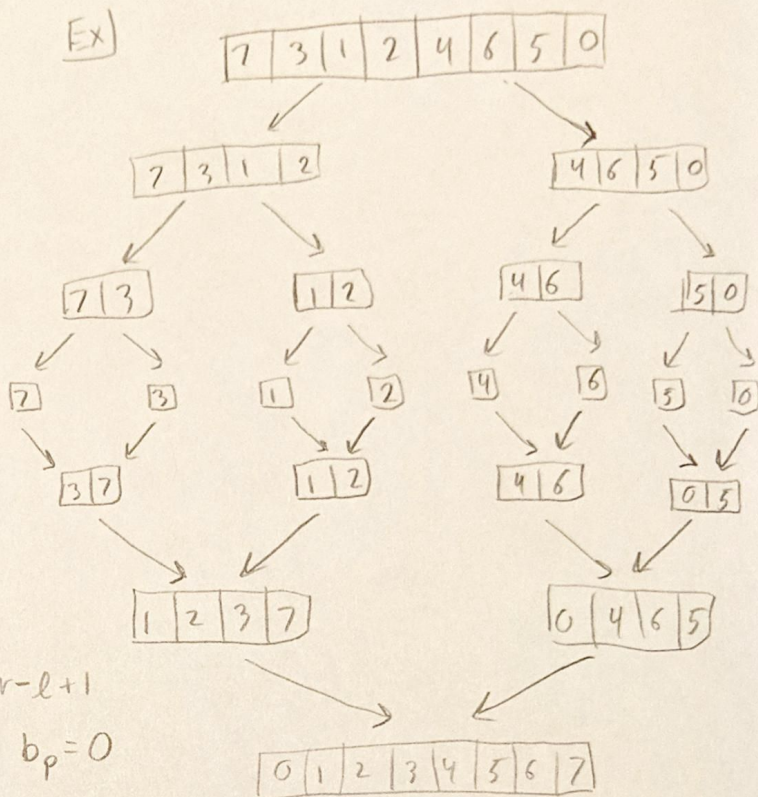
b_p ++

r_p ++

Copy B into A[l..r]

Return

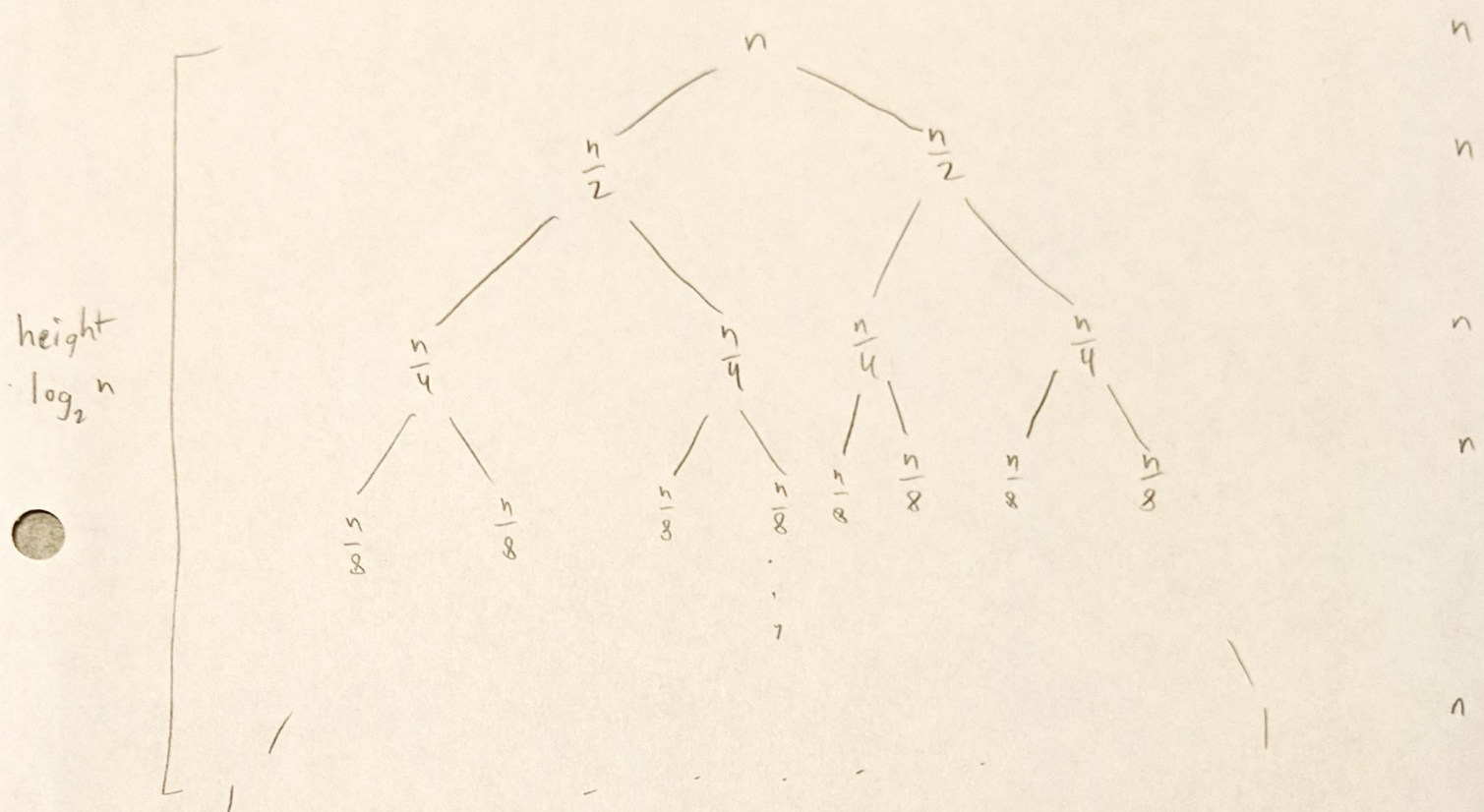
Ex)



What is the runtime of Merge Sort?

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$$

The MT immediately gives us $\Theta(n \log n)$ since $n \in \Theta(n^{\log_2 2})$.
If we want to check that with a work tree, we get



$$T(n) = \sum_{i=0}^{\log_2 n} n = (n+1) \log_2 n \in \Theta(n \log n)$$

Notice that Merge behaves, if not deterministically, then similarly enough that it always does an identical amount of work, hence $\Theta(n \log n)$ is the BC and the WC.