



Design and Analysis  
of Algorithms I

# Introduction

---

## Merge Sort (Analysis)

# Running Time of Merge Sort

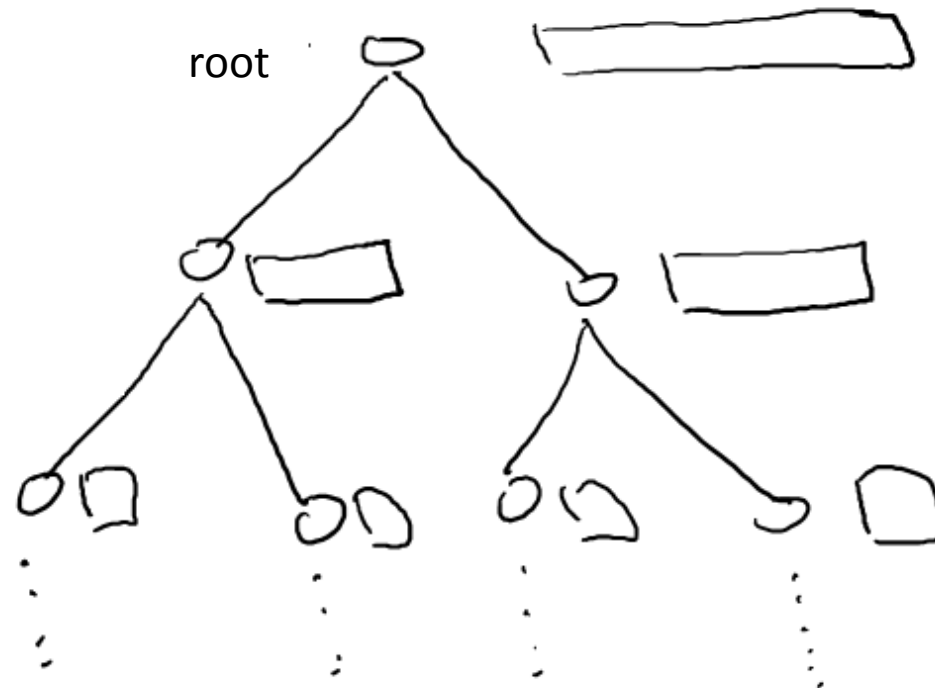
**Claim:** For every input array of  $n$  numbers, Merge Sort produces a sorted output array and uses at most  $6n \log_2 n + 6n$  operations.

# Proof of claim (assuming $n = \text{power of } 2$ ):

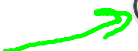
Level 0  
[outer call to  
Merge Sort]

Level 1  
(1<sup>st</sup> recursive  
calls)

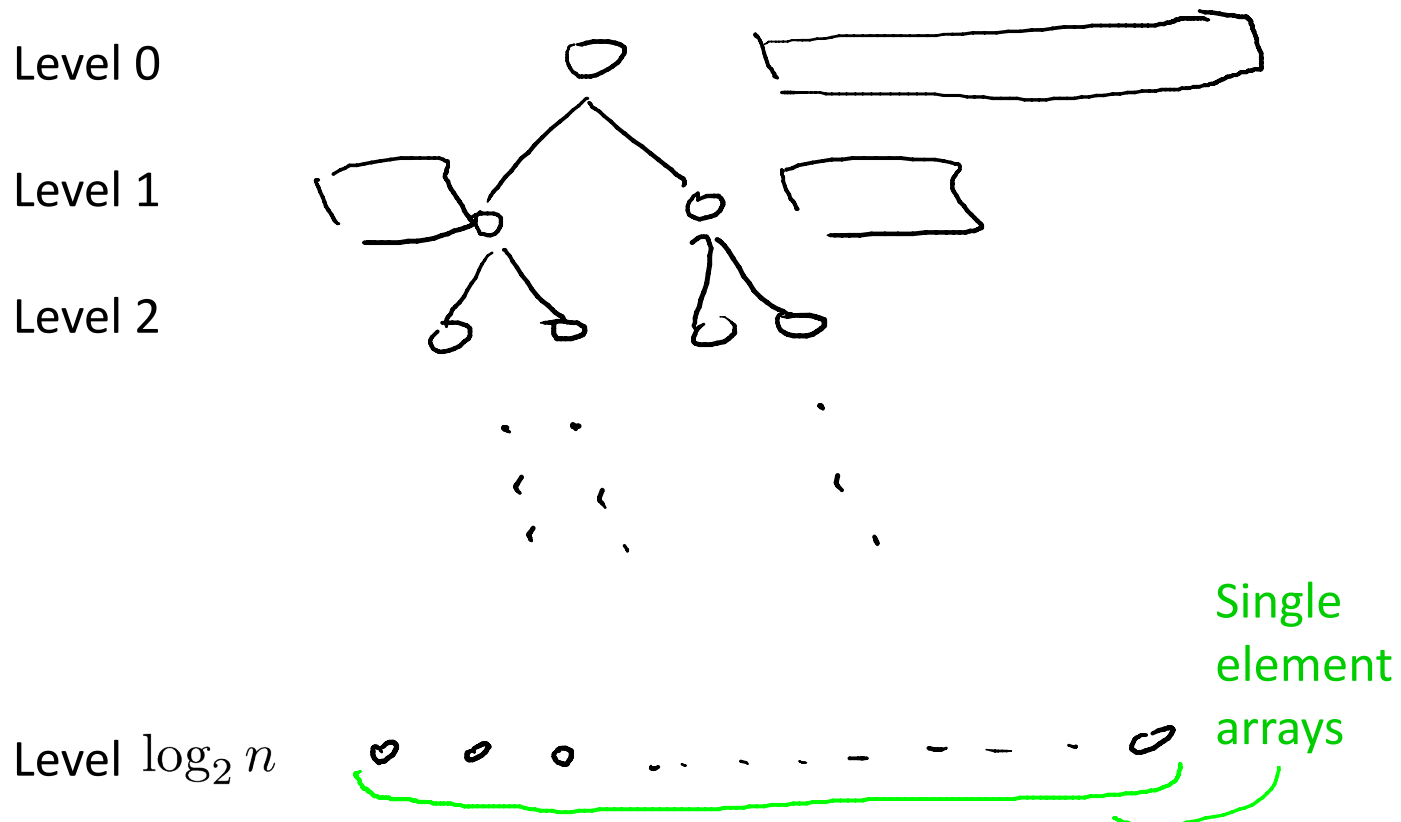
Level 2



Roughly how many levels does this recursion tree have (as a function of  $n$ , the length of the input array)?

- ☐ A constant number (independent of  $n$ ).
-  ☒  $\log_2 n$        $(\log_2 n + 1)$  to be exact!
- ☐  $\sqrt{n}$
- ☐  $n$


## Proof of claim (assuming $n = \text{power of } 2$ ):



What is the pattern ? Fill in the blanks in the following statement: at each level  $j = 0, 1, 2, \dots, \log_2 n$ , there are <blank> subproblems, each of size <blank>.

☐  $2^j$  and  $2^j$ , respectively

☐  $n/2^j$  and  $n/2^j$ , respectively

 ☐  $2^j$  and  $n/2^j$ , respectively

☐  $n/2^j$  and  $2^j$ , respectively

## Proof of claim (assuming $n = \text{power of } 2$ ) :

上一节证明过，在一个size为m的问题中，归并排序的消耗为6m，而这里可以求出那个整体就是6m

At each level  $j=0,1,2,\dots, \log_2 n$ ,

Total # of operations at level  $j = 0,1,2,\dots, \log_2 n$

$$\leq 2^j * \underline{6\left(\frac{n}{2^j}\right)} = 6n$$

# of level-j  
subproblems

Size of level-j  
subproblem

Work per level – j  
subproblem

Total

$$6n(\log_2 n + 1)$$

Work  
per level

# of  
levels

# Running Time of Merge Sort

**Claim:** For every input array of  $n$  numbers, Merge Sort produces a sorted output array and uses at most  $6n \log_2 n + 6n$  operations.

QED!