Design and Analysis of Algorithms I

Introduction

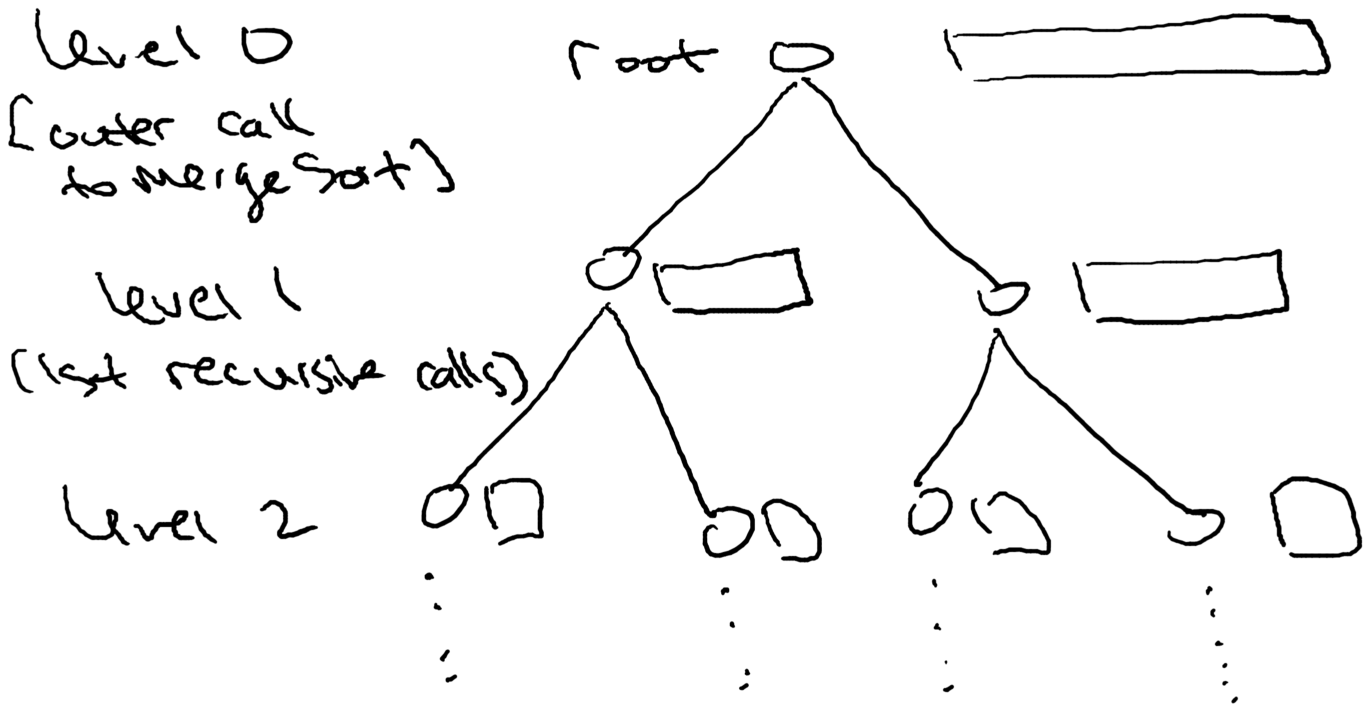
Merge Sort (Analysis)

Tim

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.

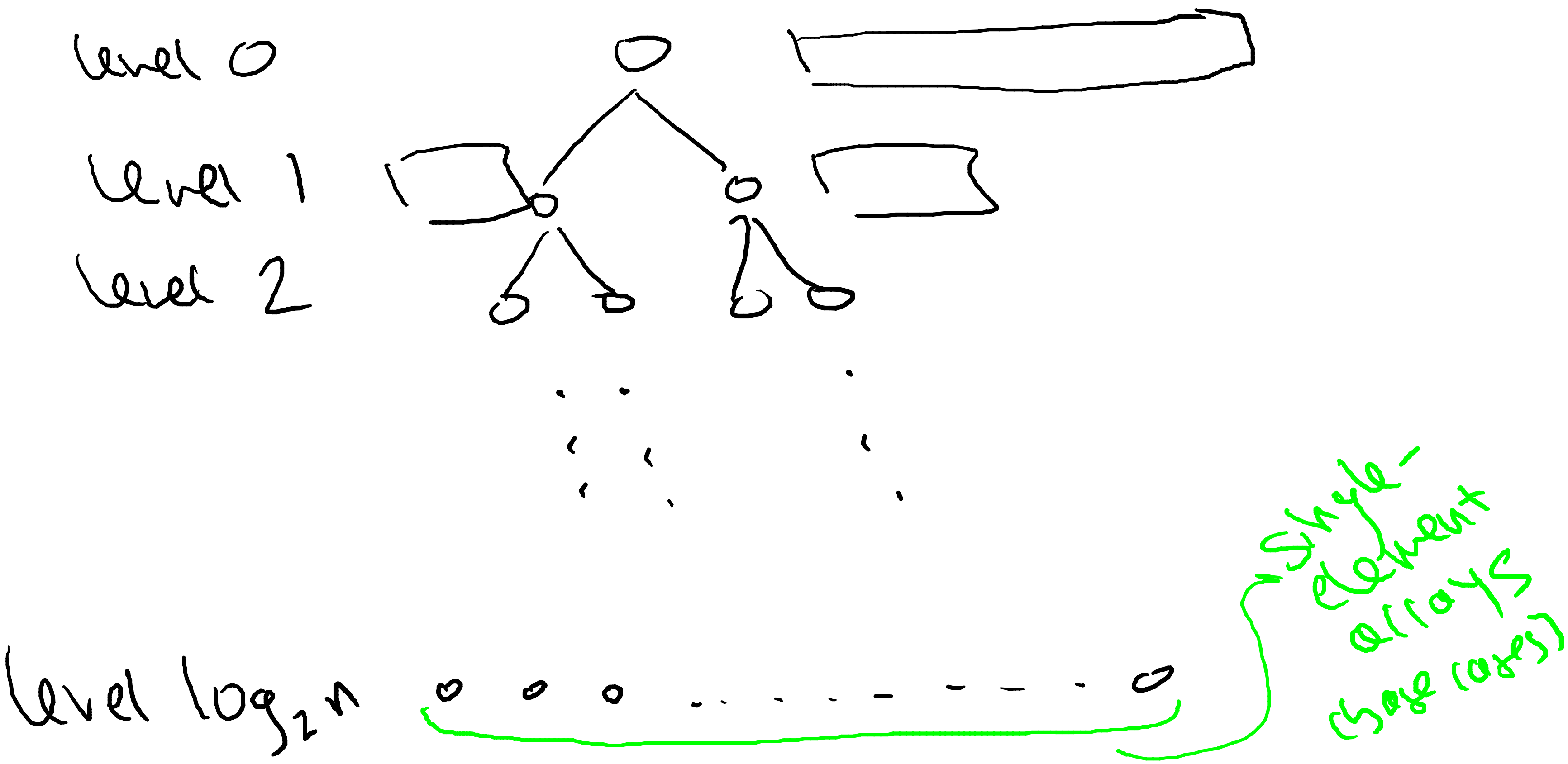
Proof of claim (assuming n = power of 2):



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 = power of 2):



What is the pattern? Fill in the blanks in the following statement: at each level j=0,1,2,…, $\log_2 n$, 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 = power of 2):

At each level j=0,1,2,..., $\log_2 n$, there are 2^j subproblems, each of size $n/2^j$.

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

$$\le \underbrace{\cancel{2^j}}_{\text{\# of level-}j\text{ subproblems}} \cdot \underbrace{6\left(\frac{n}{\cancel{2^j}}\right)}_{\text{work per level-}j\text{ subproblem}} = 6n$$

size of level-j subproblem

$$\boxed{\text{Total}}$$

$$\underbrace{6n}_{\text{work per level}}\underbrace{(\log_2 n + 1)}_{\text{\# of levels}}$$

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!