# MergeSort Analysis using recurrence tree method

Sorting: (Log$^n$) 1,2,...n indicates position

(nondecreasing)

Input: A sequence of numbers $(a_1, a_2, \dots, a_n)$

Output: A permutation (reordering)

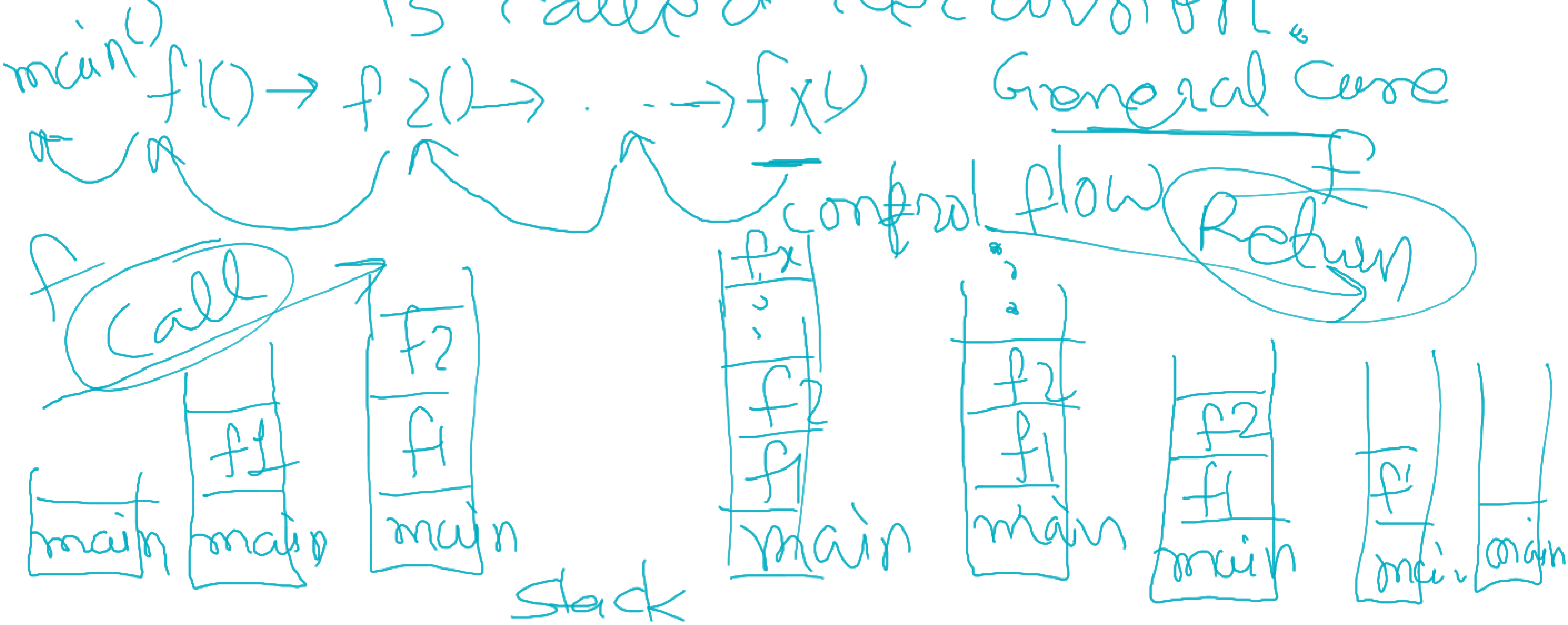
$(a_1', a_2', \dots, a_n')$ of input sequence

Such that

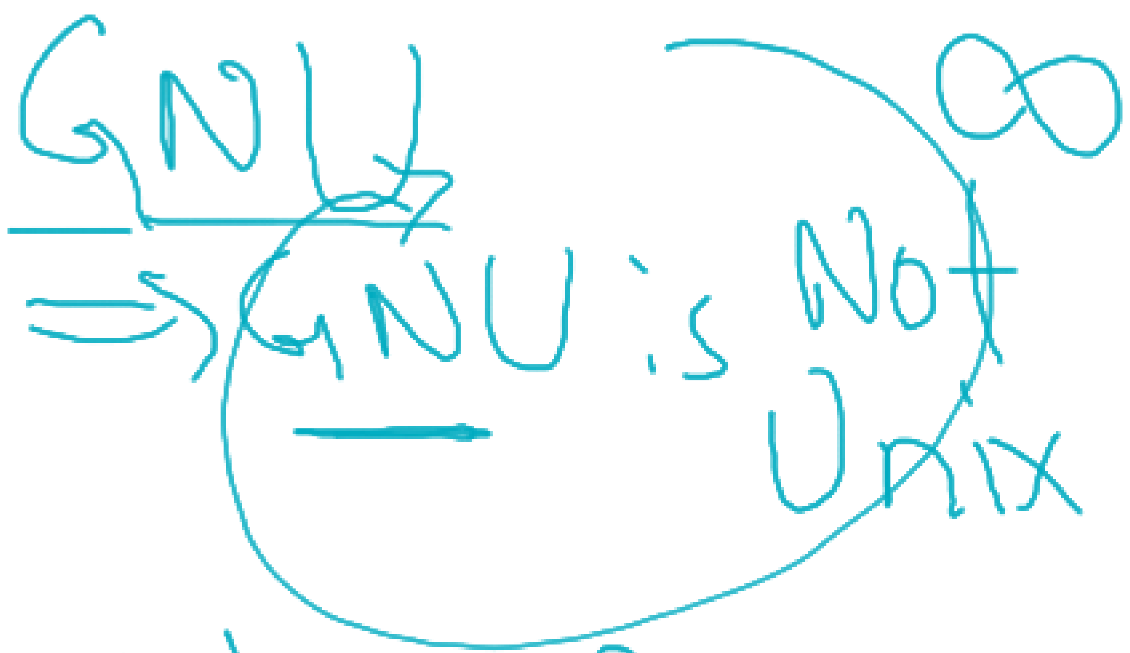
$$a_1' \le a_2' \le a_3' \le \dots \le a_n'$$

what is recursion?

If the function is calling itself, is called recursion.



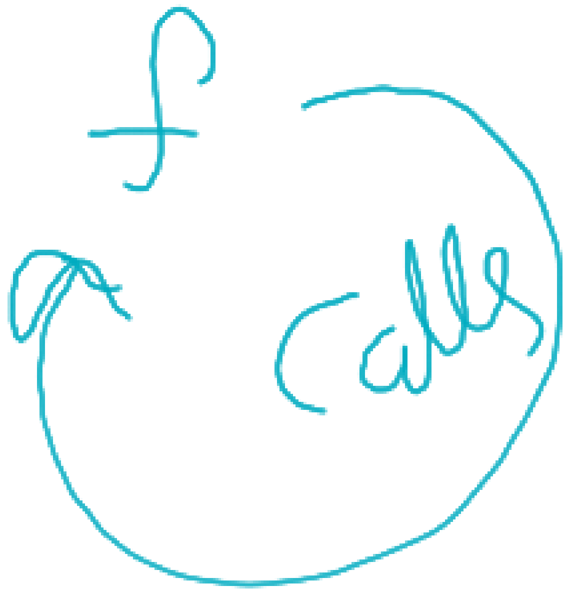
$f \to f \to f \to f \to f$



GNU

⇒ GNU is Not Unix

∞



Circle.

What is the problem?

Infinite Execution

So, recursion must have a terminating condition.

"Base Case"

Not to confuse with Worst, Best, Avg Case

Base case means the simplest scenario which you will be solving using some straight forward/Brute Force approach.

---

With respect to merge sort (Algo) what is base case?

"merging of two sorted subarrays."

Order of growth of a base case

19 25 27 35 49

Sc1 → 5

~1 compar

5 19 25 27 35 49

copy

Sc2 49    5 19 25 27 35 ↑ ~n

5 19 25 27 35 49

Copy over

Sc3 $n/2$

5 17 29 31 | 15 25 30 34 $n/2$

| comp # | 5 | 15 | 17 | 25 | 29 | 30 | 31 | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 1 | 1 | 1 | 1 | . . . | } Important |

Order of comparisons is n

To fill n positions you are bound to do comparison for merge algorithm.

FinalAnsnre

let $T(n)$ = Running Time on a Problem of size $n$

$T(n) = \Theta(1)$, if $n \le C$ Base case

$n=1$

size single element

1 meaning constant time

(It does not depend on size(N))

$= aT(n/b) + D(n) + C(n)$, otherwise

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

Later on using master theorem will be solving (order of $n \log_2 n$)

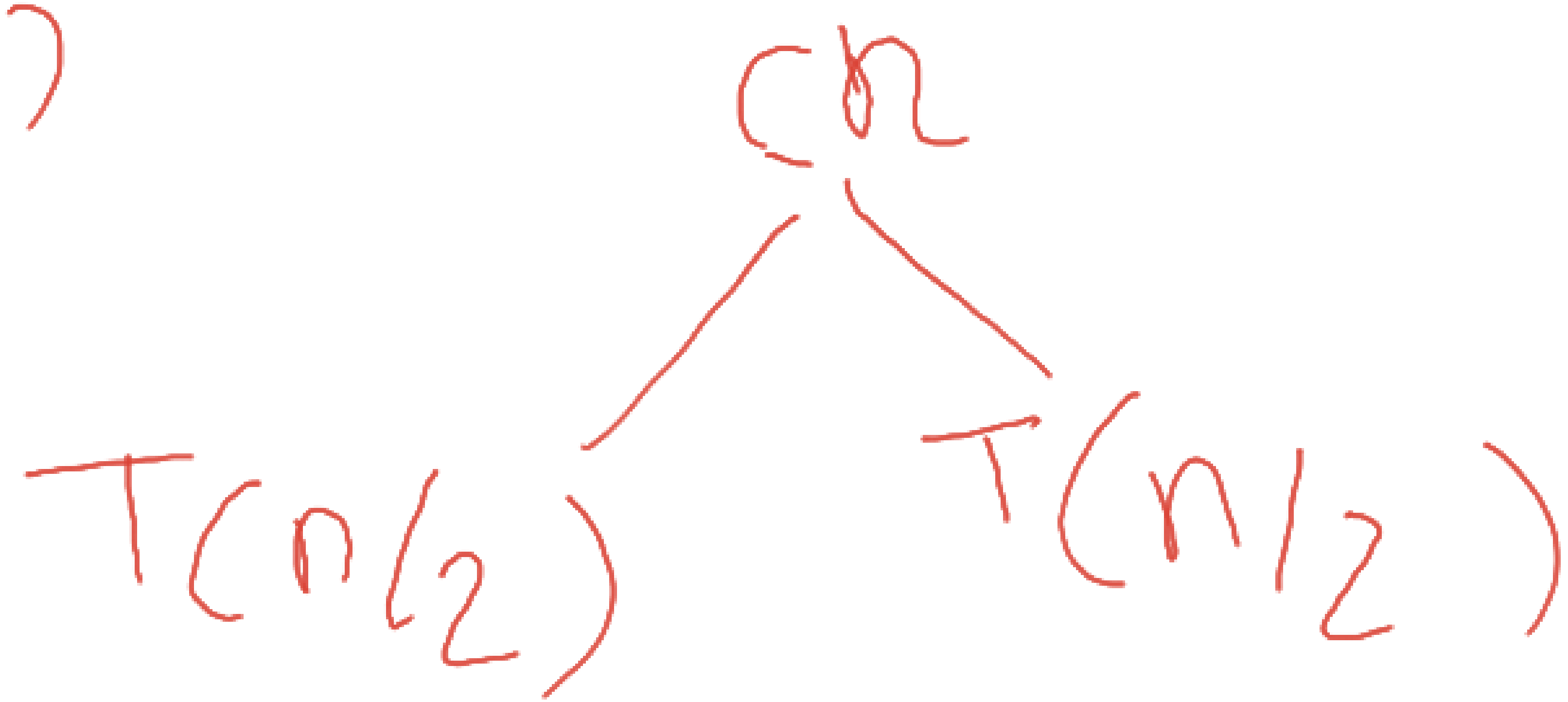
For now lets do it using recurrence tree method

Draw a recursion tree

It will show successive expansions of the recurrence.

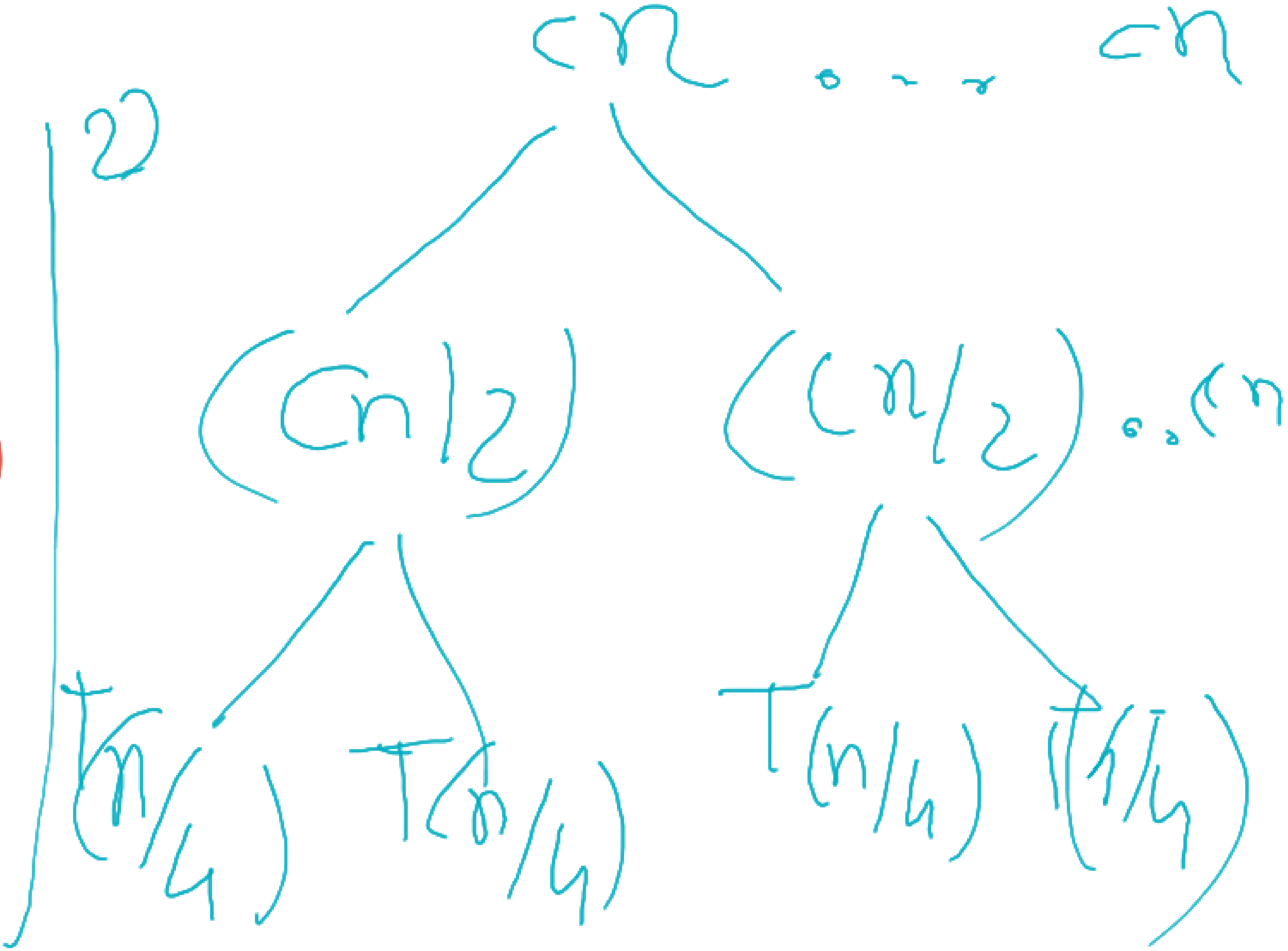
# Original Problem

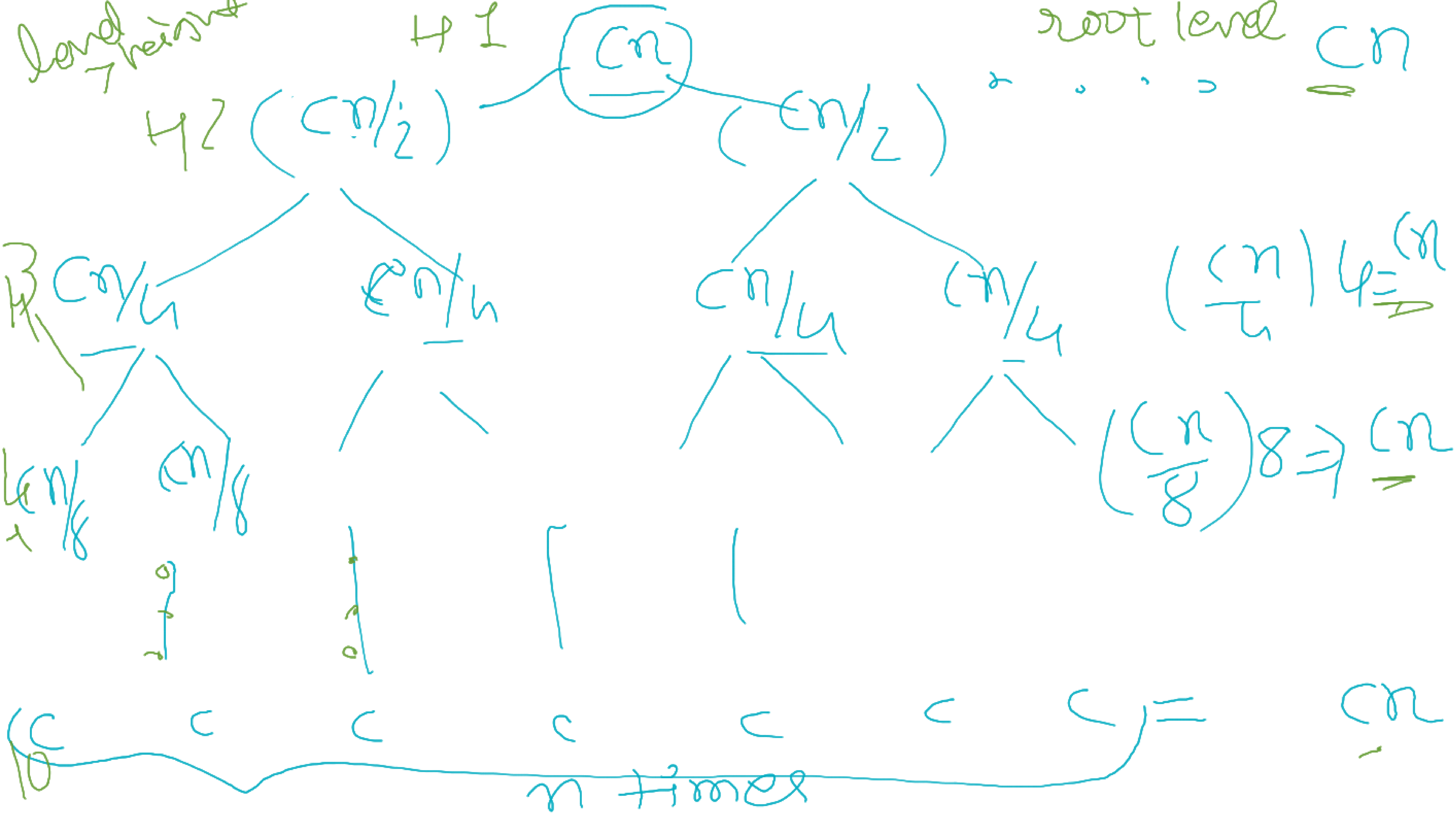
1)

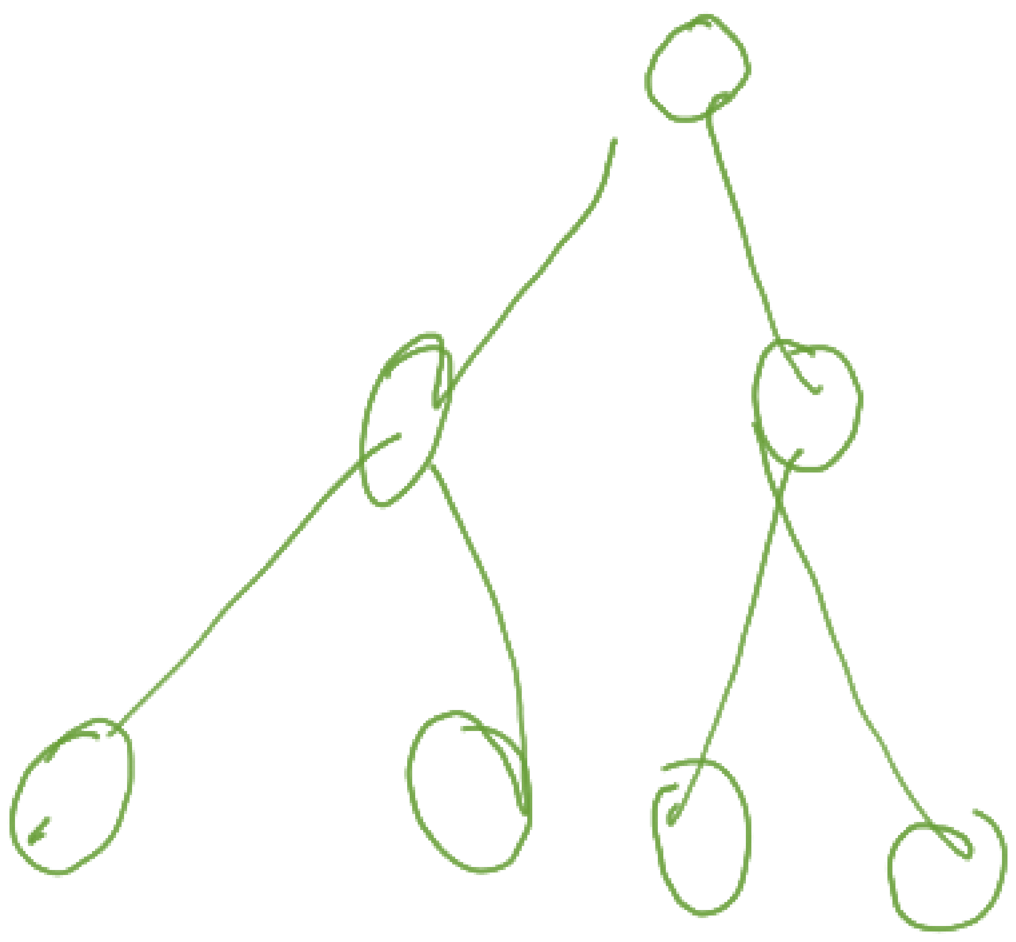


T(n)

2)




level
height
H 1
cn
root level
cn
H 2
cn/2
cn/2
3 H
cn/4
cn/4
cn/4
cn/4
(cn/4) 4 = cn
4
cn/8
cn/8
(cn/8) 8 => cn
c
c
c
c
c
c
c
= cn
n times
10



Levels 3

Height 2

Total $cn + cn + cn + \ldots + cn$ $\log_2 n$ times

$\boxed{\underline{cn \lg n}} + \underline{cn}$

1024 $2^{10}$ $\log_2 1024$ $\log_2 2^{10}$

$\rightarrow 10$

height

After removing lower order terms & constant

$$\boxed{n\log n}$$

merge sort is order growth of $n\log n$.

Assignment:

Analyse base case method partition use, to quick sort.