

Best-case / Worst-case / Avg-case.

Linear-search

(arr, target)

```
for (int i=0; i<n; i++) {
    if (arr[i] == target)
        return i;
}
```

return -1;

Best-case: $O(1)$.

Worst-case: $O(n)$.

Avg-case: $O(n)$.

$0^{th} \rightarrow 1$
 $1^{st} \rightarrow 2$
 $2^{nd} \rightarrow 3$
 \vdots
 $(n-1)^{th} \rightarrow n$
 absent $\rightarrow n+1$

$$\begin{aligned} \text{Avg} &= \frac{1+2+3+\dots+(n+1)}{n+1} \\ &= \frac{\frac{(n+1)(n+2)}{2}}{(n+1)} = \frac{n+2}{2} \end{aligned}$$

Quick Sort: Avg-case: $O(n \log n)$

Worst case: $O(n^2)$.

Space Complexity & Auxiliary Space

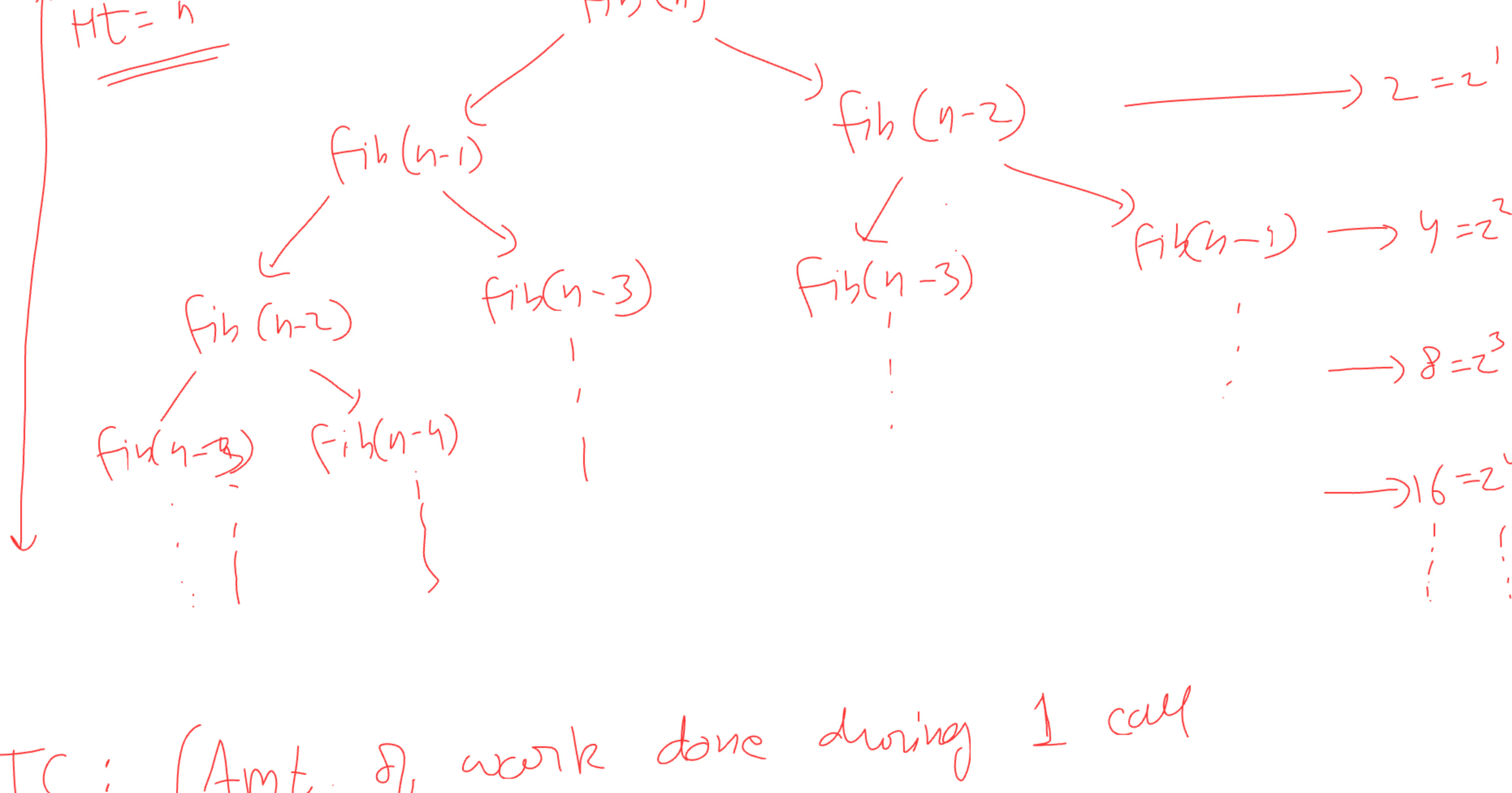
$$\left[\begin{array}{l} \text{Space-Complexity} = \text{Input space} \\ + \\ \text{Aux. space} \\ + \\ \text{Output space.} \end{array} \right] \# \text{ IMP}$$

Recursive Runtimes:

fib: 0, 1, 1, 2, 3, 5, 8, 13, 21, ...

$$\left[\begin{array}{l} \text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2) \\ (n > 1) \end{array} \right] \left\{ \begin{array}{l} \text{fib}(0) = 0 \\ \text{fib}(1) = 1 \end{array} \right.$$

```
int fib(int n) {
    if (n <= 1) return n;
    return fib(n-1) + fib(n-2);
}
```



TC: (Amt. of work done during 1 call

x Total no. of function calls)

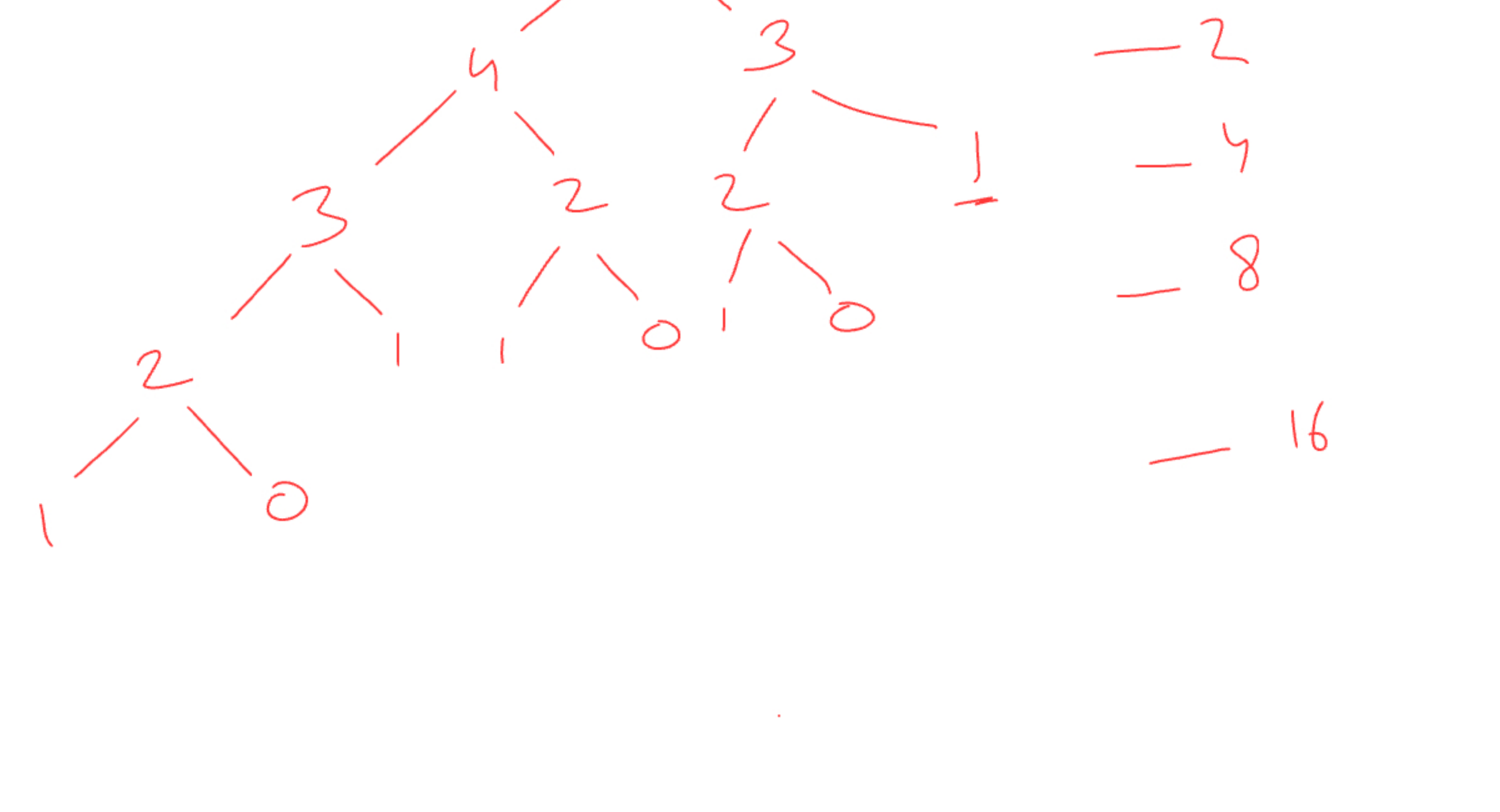
$$\text{Total no. of func calls} = 2^0 + 2^1 + 2^2 + 2^3 + \dots + 2^{n-1} = \frac{2^0(2^n - 1)}{2 - 1} = \underline{\underline{O(2^n)}}$$

Sum of AP = $\frac{a(r^n - 1)}{r - 1}$ where $a = 2^0, r = 2, n = n$

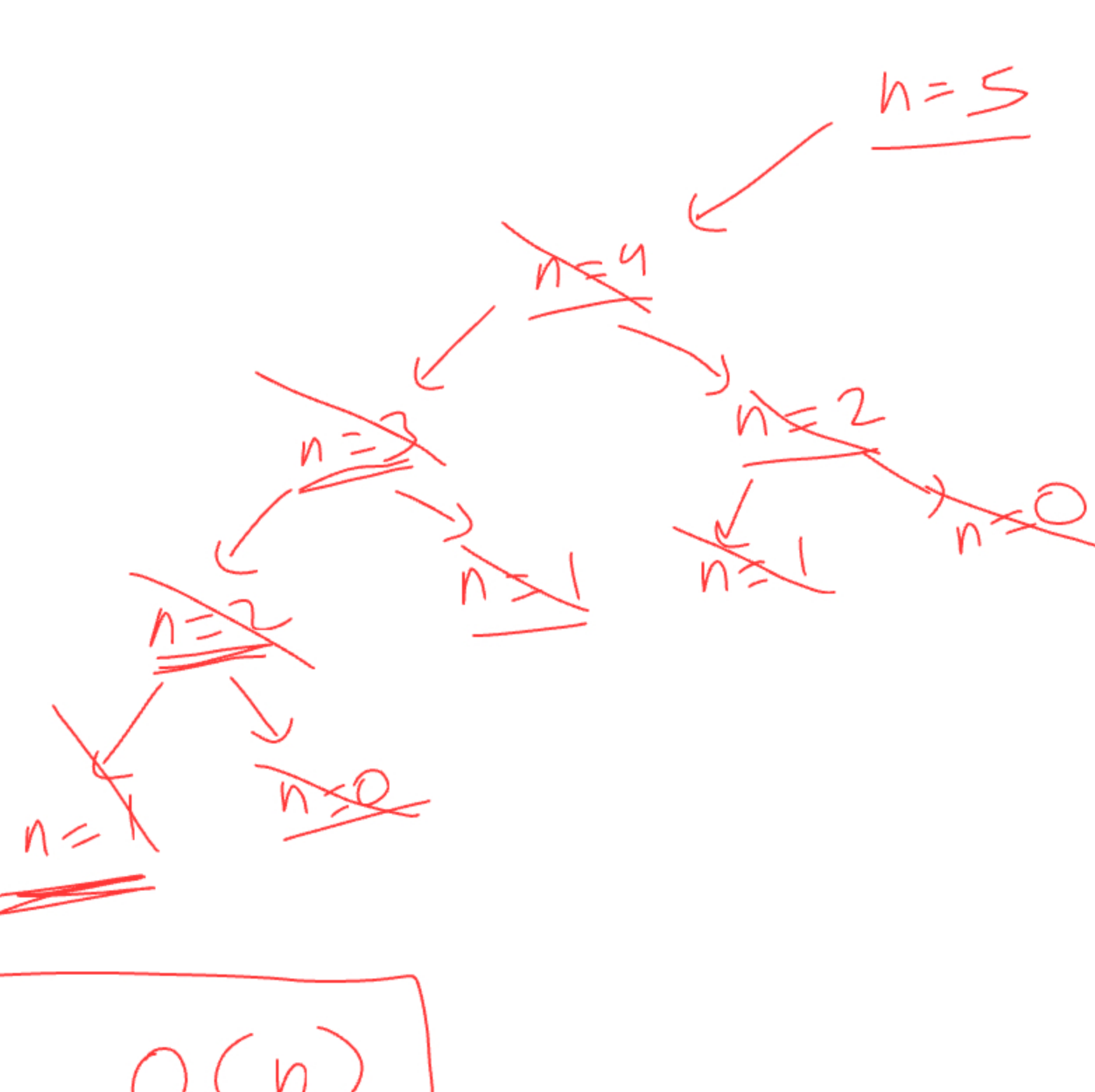
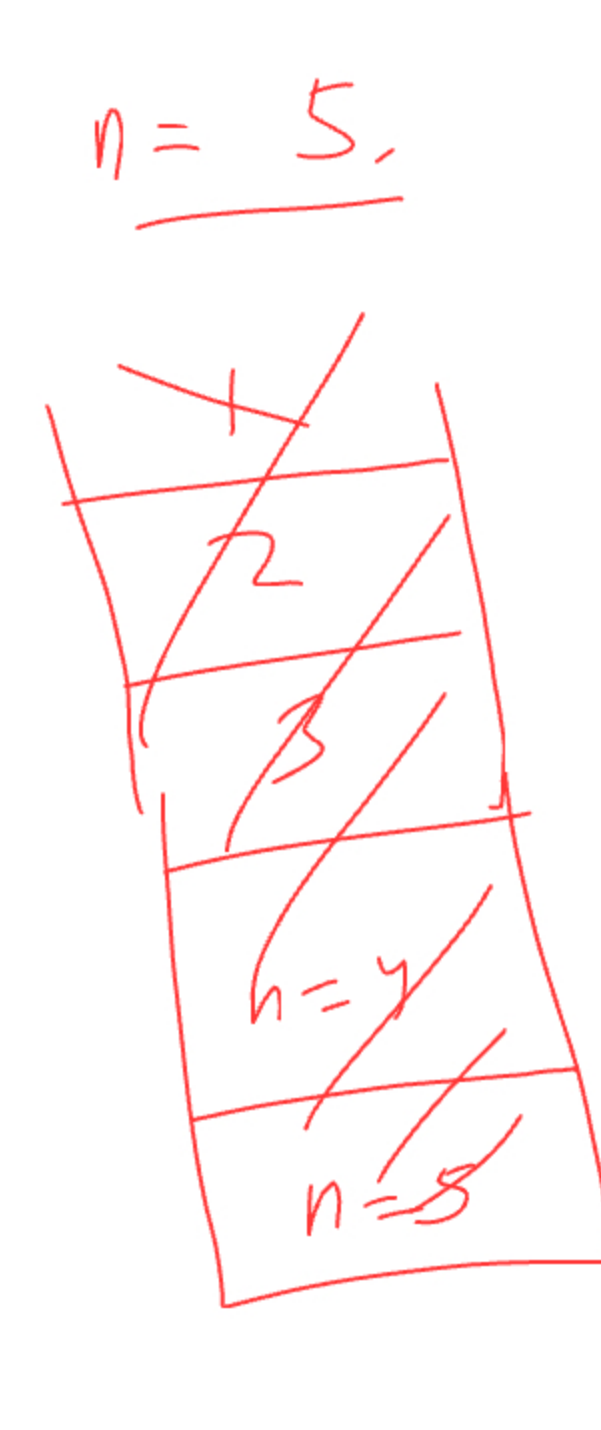
work done by 1 func call = $O(1)$

NOTE: Assuming rec. calls to be $O(1)$

$\Rightarrow \boxed{\text{TC} = O(2^n)} \#$



```
int fib(int n) {
    if (n <= 1) return n;
    return fib(n-1) + fib(n-2);
}
```



AS: $O(n)$

pass by value.

m-sized.

```
int func(int n, vector<int> arr) {
    if (n <= 1) return n;
    ... do something ...
    return func(n-1, arr) + func(n-2, arr);
}
```

n=5, arr

n=4, arr

n=3, arr

n=2, arr

AS = $O(nm)$.

pass of ref.

```
int func(int n, vector<int> &arr) {
    if (n <= 1) return n;
    ... do something ...
    return func(n-1, arr) + func(n-2, arr);
}
```

AS: $O(n)$