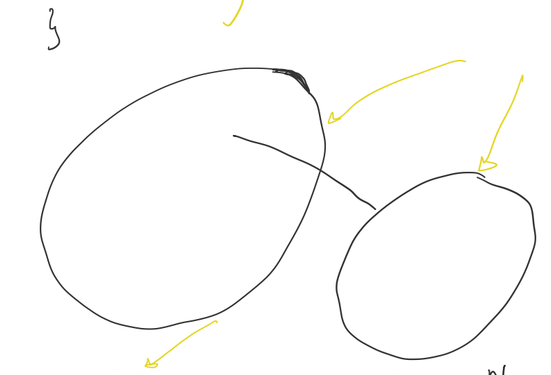
Intro to Recursion

Friday 11 August 2023 9:44 PM

Intro to Rec

→ Recursion ? = f() { g() } ✗

f() { f() }



Sum(N) = 1+2+3+4+5+ ... +N

Sum(N) = Sum(N-1) + N

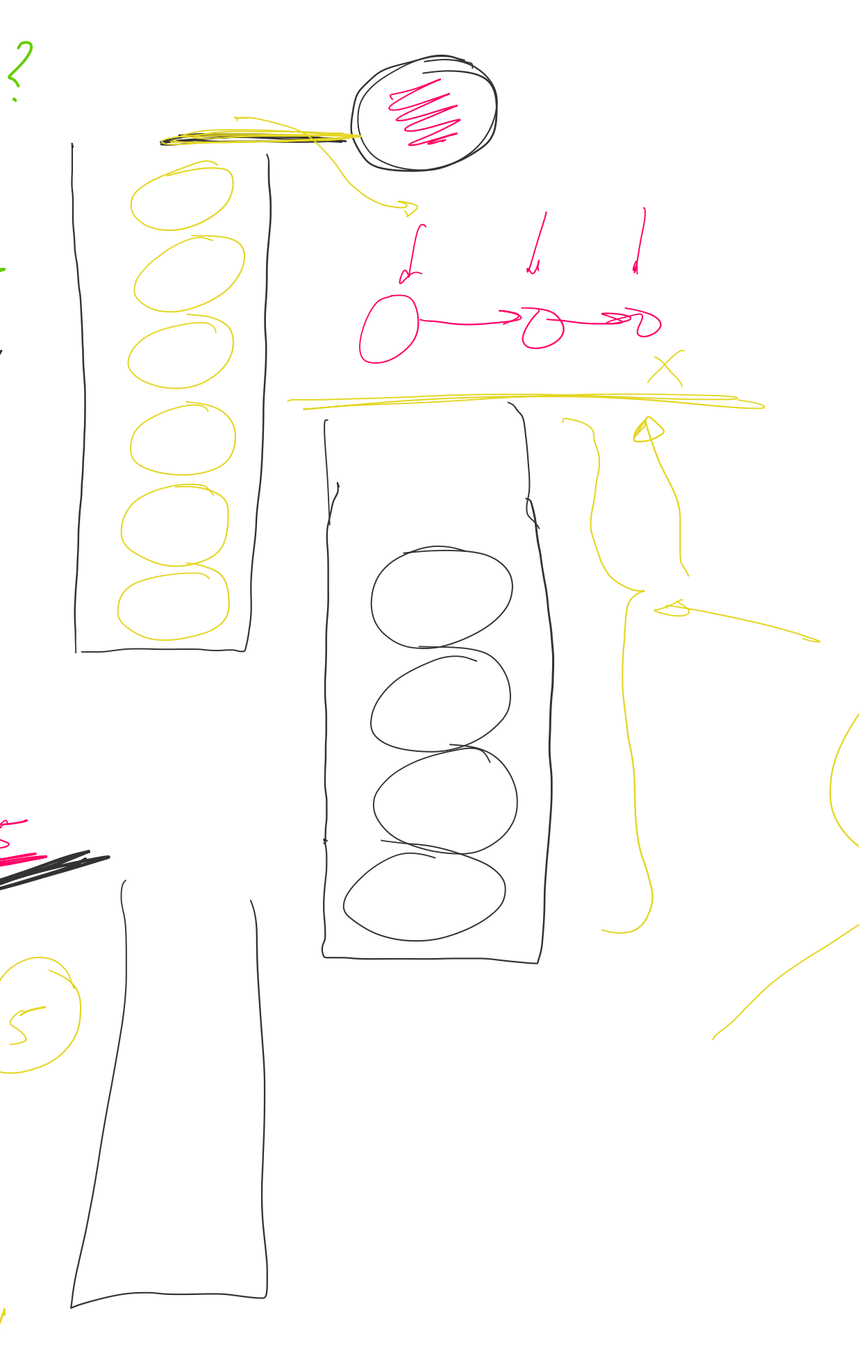
```
int sum(N) {
    if(N<=0) return 0
    return N + Sum(N-1);
}
```

Sum(0) ?

4+ 3+ 2+

N × O(1)

SC: O(N)



TL: 1 sec

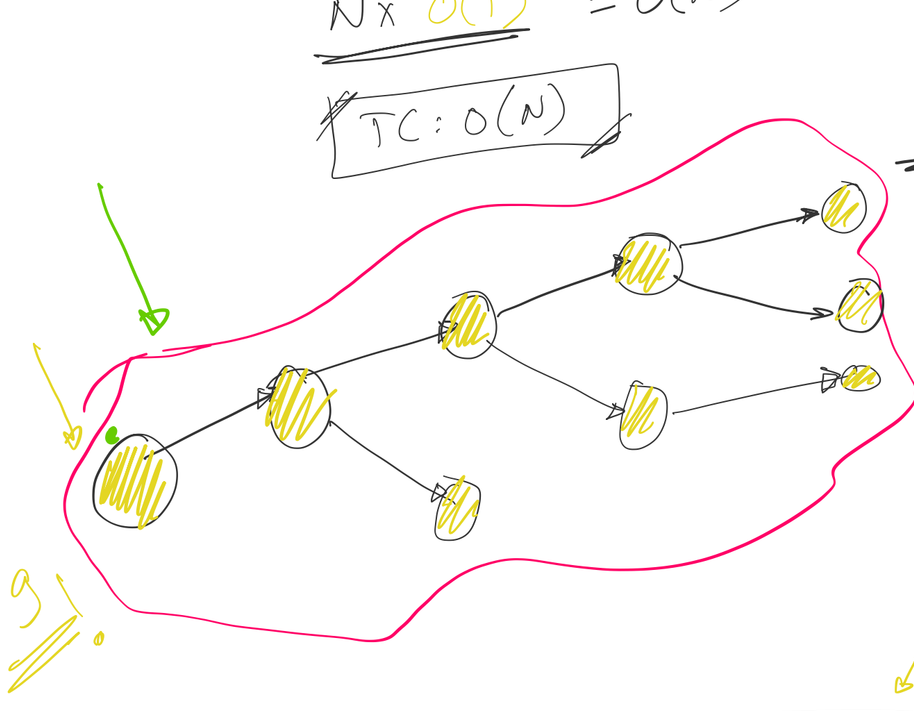
ML: 256MB/512MB

1GB

~ 10^6

N × O(1) = O(N)

TC: O(N)



SC = # max active fn calls × Space req by 1 fn call

N × O(1) = O(N)

Sum(N) → Sum(N-1) → Sum(N-2) ... → Sum(1)

N

TC = total # fn calls × time req. by 1 fn call

N × O(1)

TC: O(N)

Q A → N integers. Find the sum of array!

A: [3, 4, 5, 2, 3, 1]

Sum(N) = A[0] + A[1] + ... + A[N-1]

Sum(N) = Sum(N-1) + A[N-1]

A

```
int sum(N) {
    if(N==1) return A[0];
    return A[N-1] + sum(N-1);
}
```

N=0

S(0) +

S(-1)

S(-2)

S(-3)

[ ]

S(1)

A[-1] + ...

[1 2 3]

S(3) → S(2) → S(1) → S(0)

TC: O(N)

SC: O(N)

TC: O(N)

SC: O(1)

```
Sum = 0
for(i=0; i<n; i++) {
    Sum += A[i];
}
```

1 sec

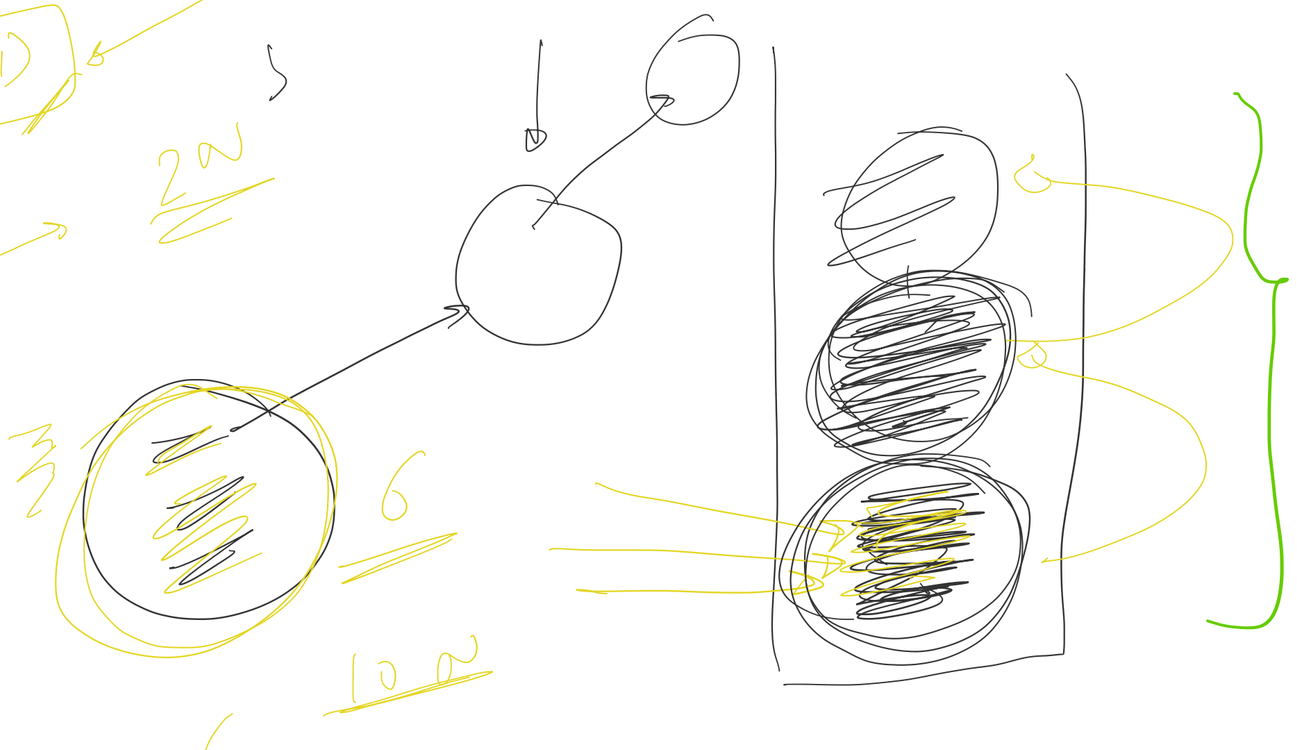
+, -, ×, / → ~10^8

2N

3

6

10N



1) Simplicity

2)

f(--) -

Q Fibonacci

1 2 3 4 5 6 7 8 9 10

0 1 1 2 3 5 8 13 21 34

TOP DOWN

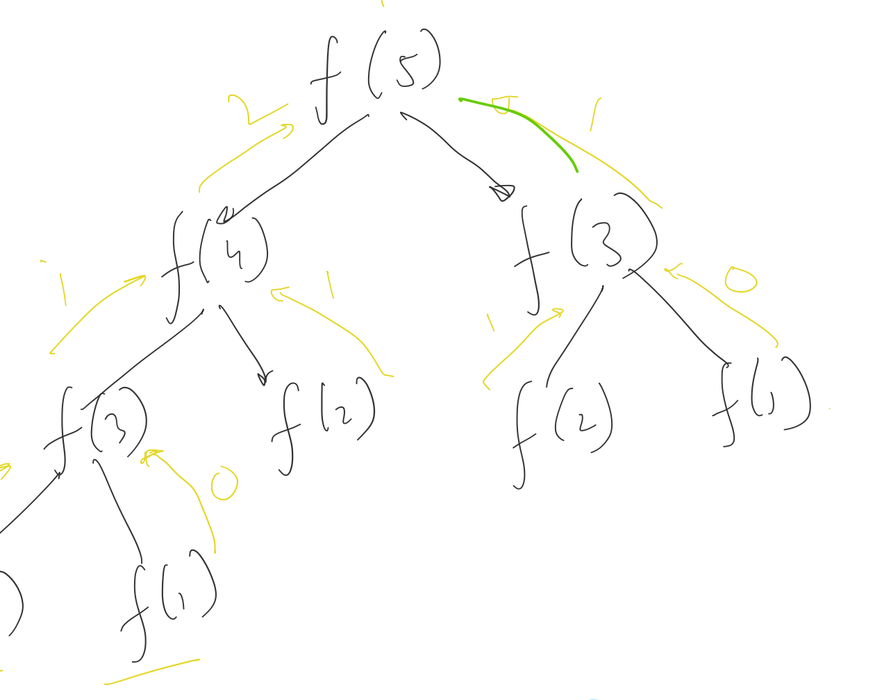
Bottom Up

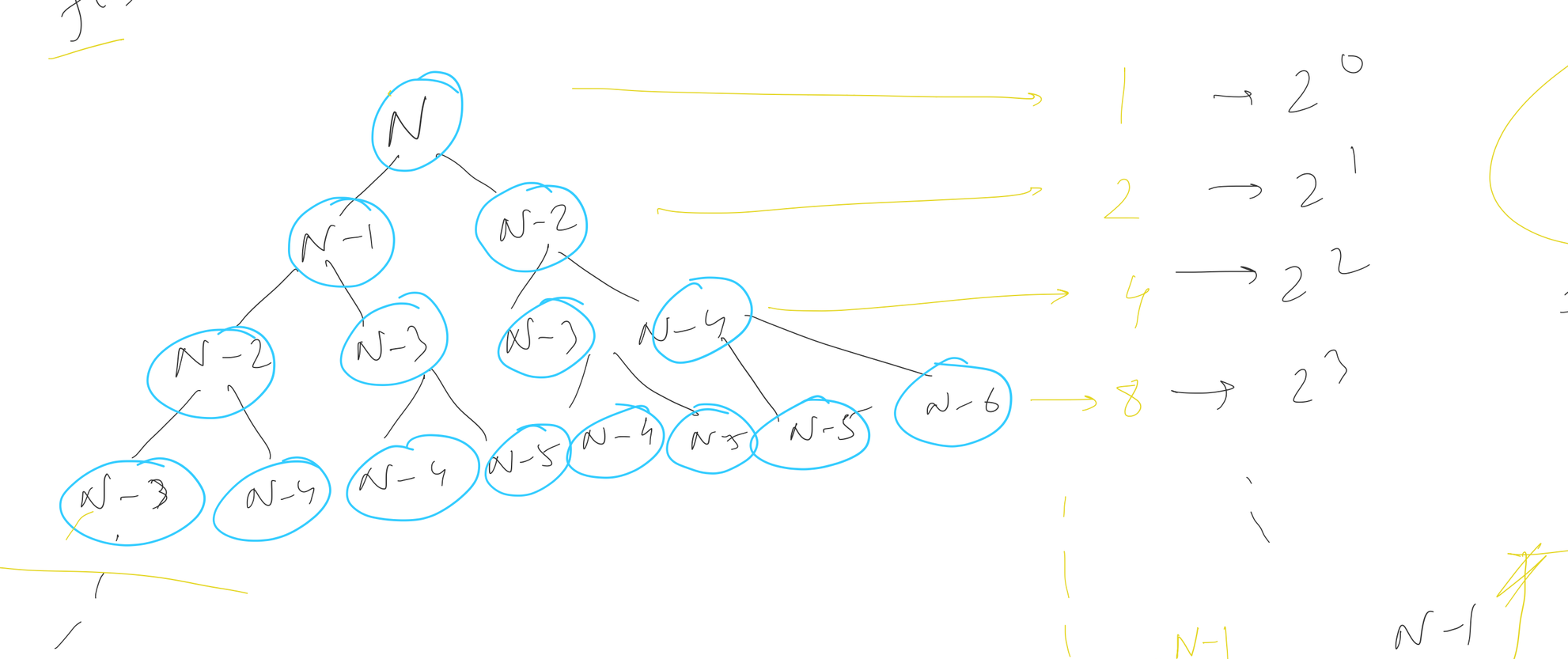
iterative soln!

f(N) = f(N-1) + f(N-2)

N ?

```
int fib(N) {
    if(N==1) return 0;
    if(N==2) return 1;
    return fib(N-1) + fib(N-2);
}
```


f(5)
f(4)
f(3)
f(3)
f(2)
f(2)
f(1)
f(2)
f(1)


N
N-1
N-2
N-2
N-3
N-3
N-4
N-3
N-4
N-4
N-5
N-4
N-5
N-5
N-6
N
1

1 → 2^0

2 → 2^1

4 → 2^2

8 → 2^3

→ 2^{N-1}

$2^0 + 2^1 + 2^2 + \dots + 2^{N-1}$

$= 2^N - 1$!

a, ar, ar², ... ar^{n-1}

$S_n = \frac{a[r^n - 1]}{r-1}$

$\frac{[2^n - 1]}{2-1}$

fn calls = $O(2^n)$

TC: $2^N \times O(1)$

TC: $O(2^N)$

N = 20 → $2^{20} \sim 10^6$

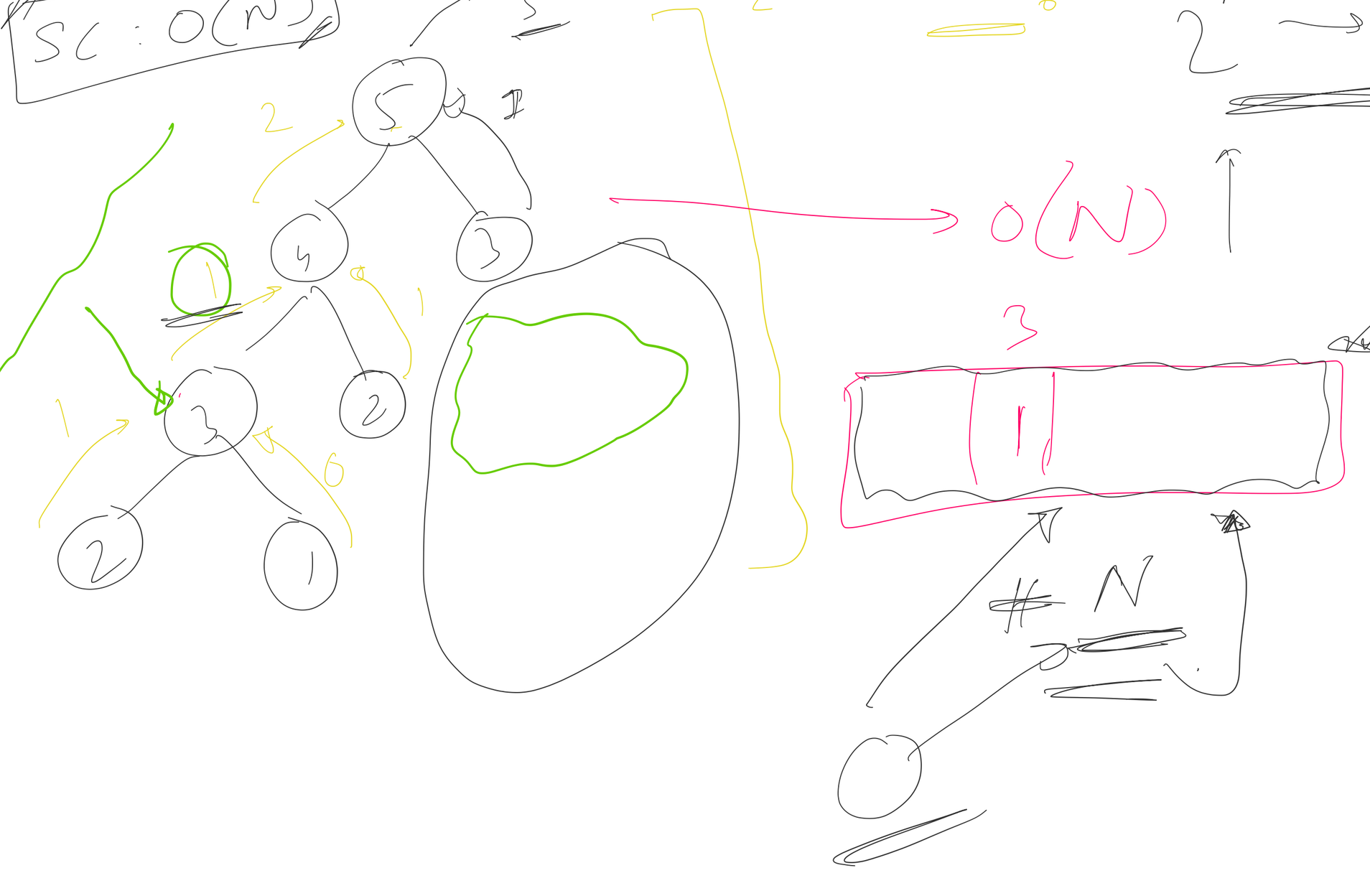
N = 30 → $2^{30} \sim 10^9$!

SC: N × O(1)

SC: O(N)

TC

$2^N \rightarrow N$


5
4
3
3
2
2
1
O(N)
N

Q Given a string, check if it's a palindrome or not!

nitin ✓ eye ✓ malayalam ✓ papa × didi × s1s ✓

appa ✓ dad ✓ mom ✓

malayalam

alayala

S

```
bool isP(st, end) {
    if(st >= end) return true;
    if(s[st] == s[end] && isP(st+1, end-1))
        return true;
    else
        return false;
}
```

st end

0 n-1

isP(0, n-1)

isP(1, n-2)

isP(2, n-3)

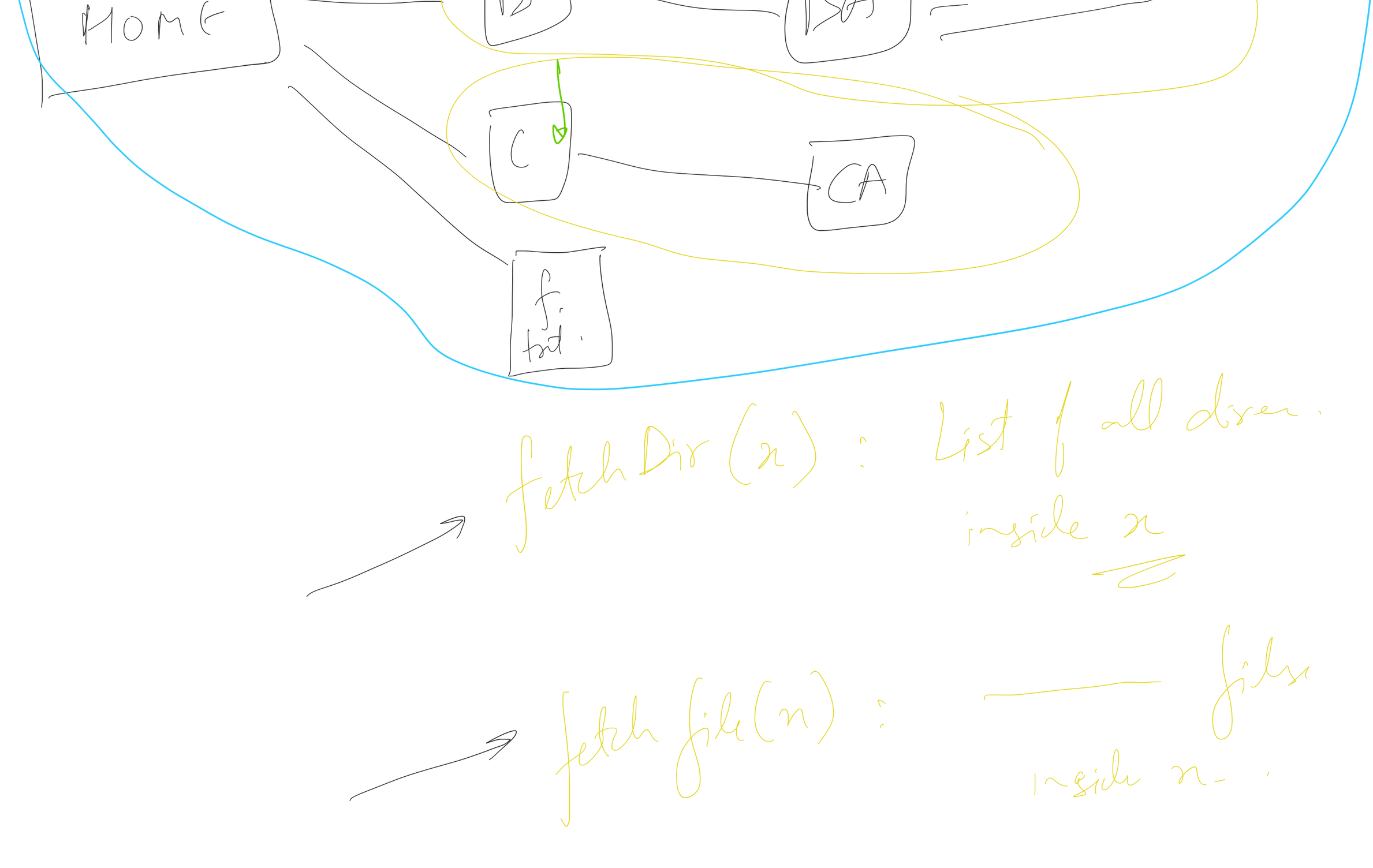
isP(3, n-4)

N/2

SC: O(N)

TC: O(N)

Q


HOME
A
B
C
f.txt
AA
AD
BA
CA
Scaler.txt
x.cpp
x.h

fetchDir(x) : List of all dirs. inside x

fetchFile(x) : ___ files inside x.

```
bool isFilePresent(directory, fileName) {
    if(directory == NULL)
        return false.

    c = fetchFiles(directory);
    if( isPresnt(C, fileName))
        true;

    d = fetchDirectory(directory);

    for(auto dir: d) {
        if(isFilePresent(dir, filename)) {
            return true;
        }
    }
    return false;
}
```