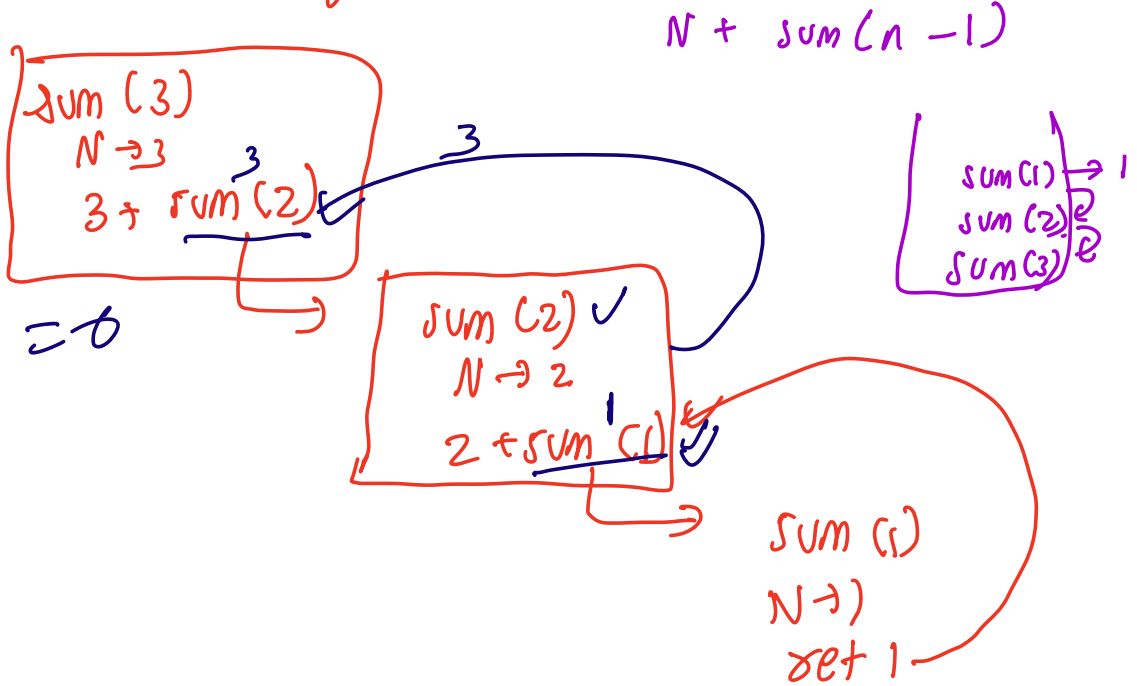
Recursion & Sorting



0 ⇒ fact(N)

fact(3) = 1 * 2 * 3

fact(4) = 1 * 2 * 3 * 4 = f(3) * 4

fact(5) = 1 * 2 * 3 * 4 * 5 = f(4) * 5

fact(n) = n * fact(n-1)

```
if (N == 0)
    return 1;
```

N

fact(N) → O(1)

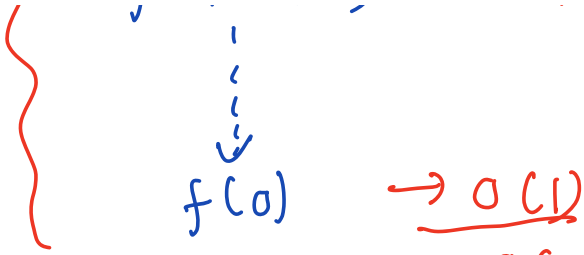
↓

fact(N-1) → O(1)

↓

fact(N-2) → O(1)

n-1 * f(N-2)



NO(1)

O(N)

O ⇒ fib(N)

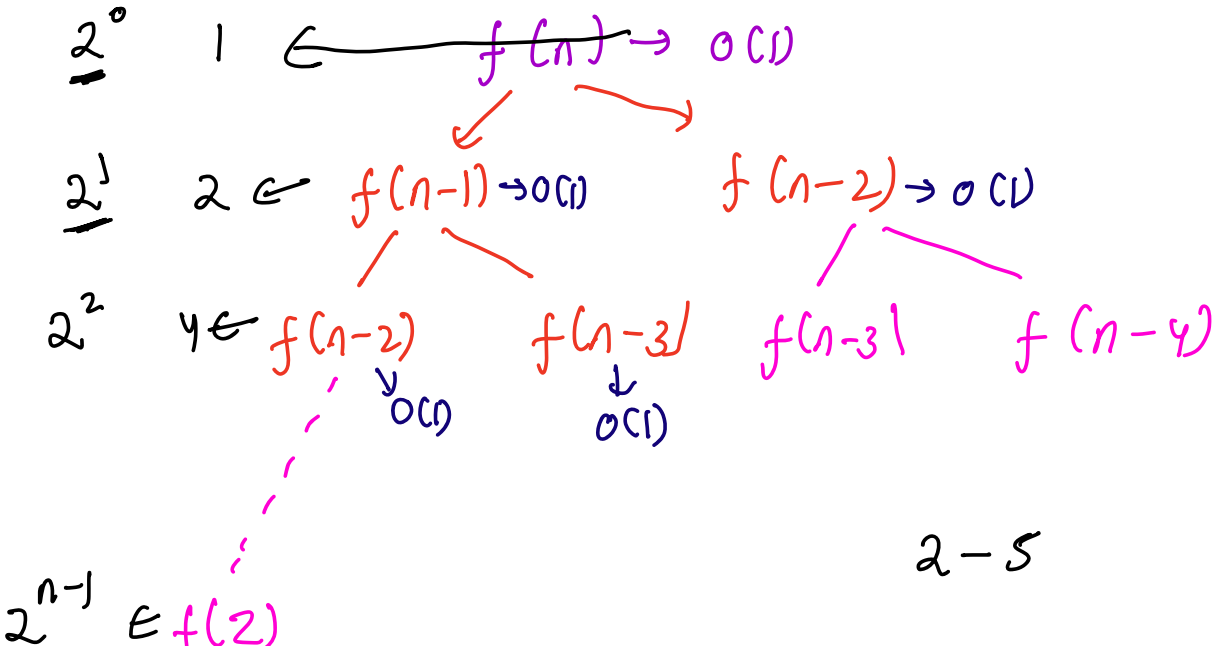
1	2	3	4	5	6
1	1	2	3	5	8

fib(4) = 1 1 2 3

fib(5) = 1 1 2 3 5

fib(6) = 1 1 2 3 5 8

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



$2^0 + 2^1 + 2^2 + \cdots + 2^{n-1}$

$a = 1$

$r = 2$

$0 \to n-1$

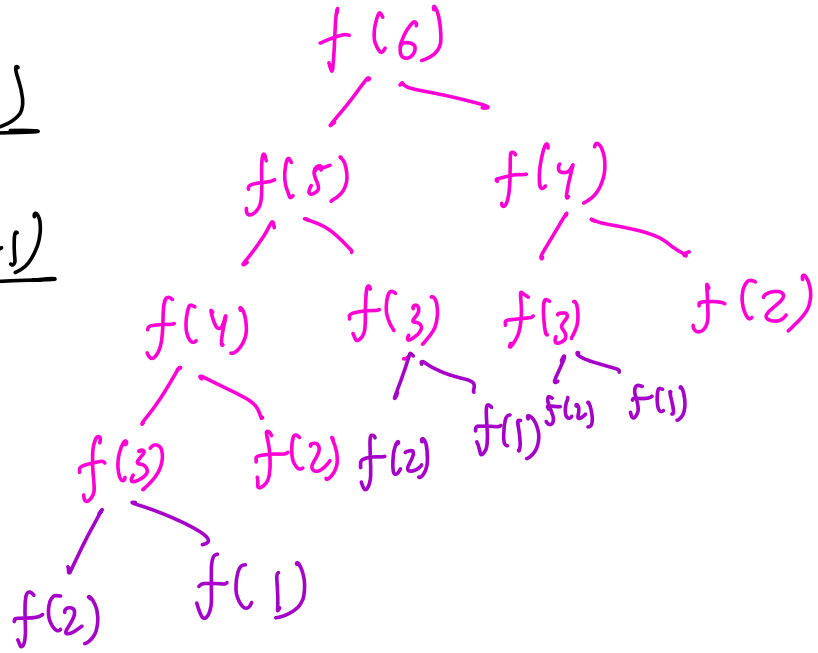
$$\frac{a(r^n - 1)}{r-1}$$

$$= \frac{1(2^n - 1)}{2-1}$$

$$= 2^n - 1$$

$$\downarrow$$

$$O(2^n)$$



TC of recursive fn = Total calls in recursive tree * TC of each func

$$f(N) = f(N-1) * N$$

$$T(N) \quad T(N-1)$$

$$T(N) = T(N-1) + T(1)$$

$$\hookrightarrow T(N-1-1) + T(1)$$

$$T(N-2) + T(1)$$

$$T(N) = T(n-2) + T(1) + T(1)$$

$$= T(n-2) + 2T(1)$$

$$\hookrightarrow T(n-2) = T(n-3) + T(1)$$

$= T(n-3) + T(1) + 2T(1)$

$= T(n-3) + 3T(1)$

$\vdots$

$T(k) = T(n-k) + kT(1)$

$T(1)$

$n-k = 1$

$k = n-1$

$= T(1) + (n-1)T(1)$

$= T(1)[1 + (n-1)]$

$= T(1)\,n$

$O(n)$

10 n

$n * n = n^2$

Google

Given a, N & P

implement pow(a, N, p) $\rightarrow$ $a^N \,\%\, p$

$5 \,\%\, 2 = 1$

$9 \,\%\, 5 = 4$

$(a+b) \,\%\, m = (a \,\%\, m + b \,\%\, m) \,\%\, m$

$(a * b) \,\%\, m = (a \,\%\, m * b \,\%\, m) \,\%\, m$

$(3)^2 = 3 * 3^1$

$= 3 * 3 * 3^0$ (1) ^{30}

$(1)^{47}$

$a=6$, $b=2$, $m=4$

$(6+2)\%4$

$8\%4$

0

$(6\%4 + 2\%4)\%4$

$2 + 2 = 4\%4$

$= 0$

$$(a^N)\%p = (a * a^{N-1})\%p$$
$$= (a\%p * a^{N-1}\%p)\%p$$

```
int pow( a, N, p)
{
    if( a==1 || N==0)
        ret 1;
    ret ( a%p * pow(a, N-1, p) )%p;
}
```

O(1) · T(n-1)

$T(N) = T(n-1) + T(1)$

$O(n)$

$a^4 = a * a^3$

$a * a * a^2$

$a * a * a * a^1$

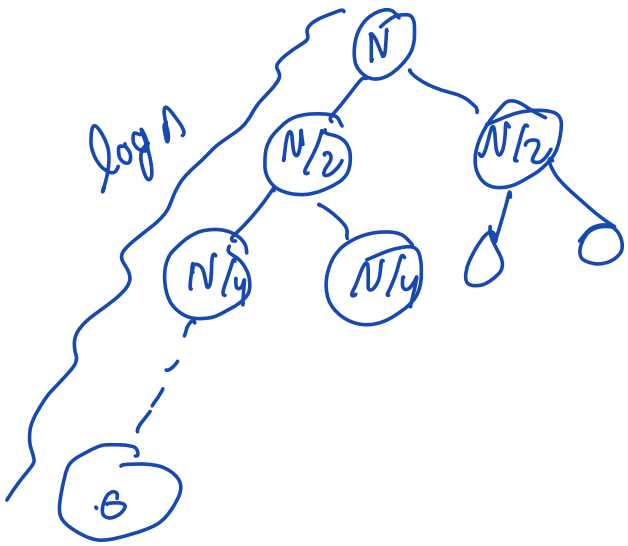
$a * a * a * a * a^0$

$a^4 = a^2 * a^2$

$= a \cdot a * a \cdot a$

$a^5 = a^2 * a^2 * a$

$a^{13} = a^6 * a^6 * a$




```
int   pow (a, N, b)
{
    if (N == 0  ||  a == 1)
        ret 1
    if (N%2 == 0)
        ret ( pow(a, N/2, b) * pow(a, N/2, b)) %b
    else
        ret ((pow(a, N/2, b) * pow(a, N/2, b))%b
             *  a%b)%b
}
```

$T(n) = T(n/2) + T(n/2) + T(1)$

$T(n) = 2\,T(n/2) + T(1)$

$$T(n/2) = 2T(n/4) + T(1)$$

$$= 2\left[2T\left(\frac{n}{4}\right) + T(1)\right] + T(1)$$

$$= 4T\left(\frac{n}{4}\right) + 3T(1)$$

$$T(n/4) = 2T(n/8) + T(1)$$

$$= 4\left[2T\left(\frac{n}{8}\right) + T(1)\right] + T(1)$$

$$= 8T\left(\frac{n}{8}\right) + 7T(1)$$

$$\vdots$$

$$T(N) = 2^k T\left(\frac{n}{2^k}\right) + (2^k - 1)T(1)$$

$$\frac{N}{2^k} = 1$$

$$N = 2^k$$

$$k = \log_2 N$$

$$2^{\log_2 N} T(1) + (2^{\log_2 N} - 1)T(1)$$

$$2^{\log_2 x} = x$$

$$N\,T(1) + (N-1)\,T(1)$$

$$T(1)\,[N + N - 1)$$

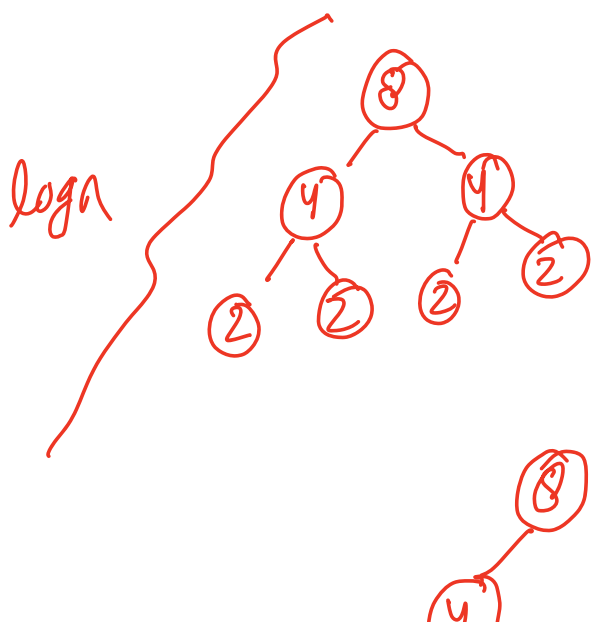
T(1) [2N-1]

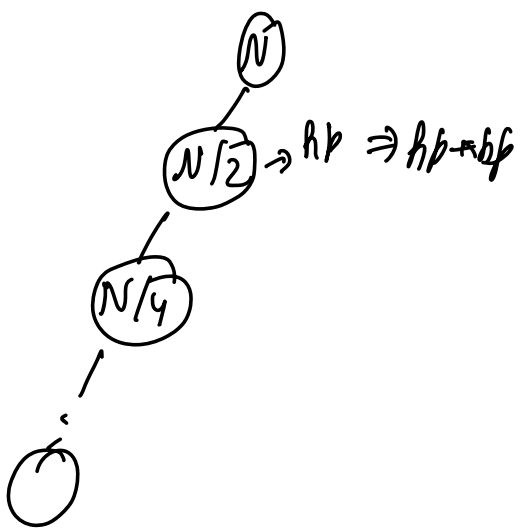
O(n)

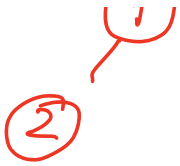
```
int pow (a, N, p)
{
    if (N == 0 || a == 1) ret 1
        hp = pow( a, N/2, p);
    if (N%2 == 0)
        ret (hp * hp) %p
    else
        ret (hp * hp) %p * a %p) %p;
}
```

$$T(n) = T(n/2) + T(1)$$

Break: 10:53







$$T(n) = T(n/2) + T(1)$$

$\hookrightarrow T(n/2) = T(n/4) + T(1)$

$$= T(n/4) + T(1) + T(1)$$

$$= T(n/4) + 2T(1)$$

$\hookrightarrow T(n/4) = T(n/8) + T(1)$

$$= T(n/8) + 3T(1)$$

$$\vdots$$

$$= T(n/2^k) + kT(1)$$

$$n/2^k = 1$$

$$n = 2^k$$

$$k = \log_2 n$$

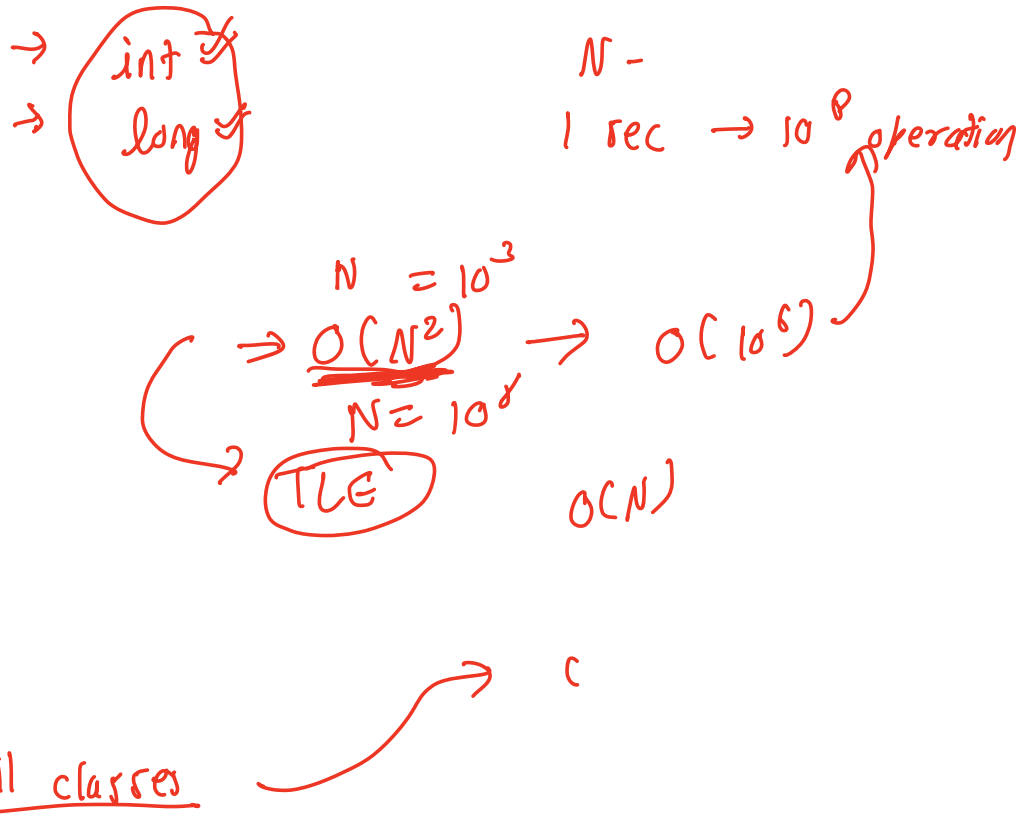
$$T(n/\log_2 n) + \log_2 n \; T(1)$$

$$T\left(\frac{n}{n}\right) + \log_2 n \; T(1)$$

$$T(1) + \log_2 n \; T(1)$$

$$T(1)[1 + \log_2 n]$$

$$O(\log_2 N)$$



All classes → C