n = 5

$5! = 5 \times 4 \times 3 \times 2 \times 1 = 1 \times 2 \times 3 \times 4 \times 5$

120

fact = 0

loop { 1 to 5

for(i=1; i<=n; i++) { fact = fact * i; ;

i <= 5

i = 1

fact = 0

fact = fact * i;

fact = 0 × 1 = 0

i = 2

fact = 1 ;

fact = 1 × 1

fact = 1

0 × 1

2nd time {

fact = 1 × 2

= 2

factorial(5) = 5 × 4 × 3 × 2 × 1

fact(4) = 4 × 3 × 2 × 1

fact(3) = 3 × 2 × 1

fact(2) = 2 × 1

fact(1) = 1

fact(0) = 1

fact(6) = 6 × fact(5)

= 6 × 5 × 4 × 3 × 2 × 1

= 6 × fact(5)

n = 6

fact(n) = n × fact(n-1);

fact(6) = 6 × fact(5);

fact(5) = 5 × fact(4)

```
fact(6) {
  return 6 x fact(5)
```

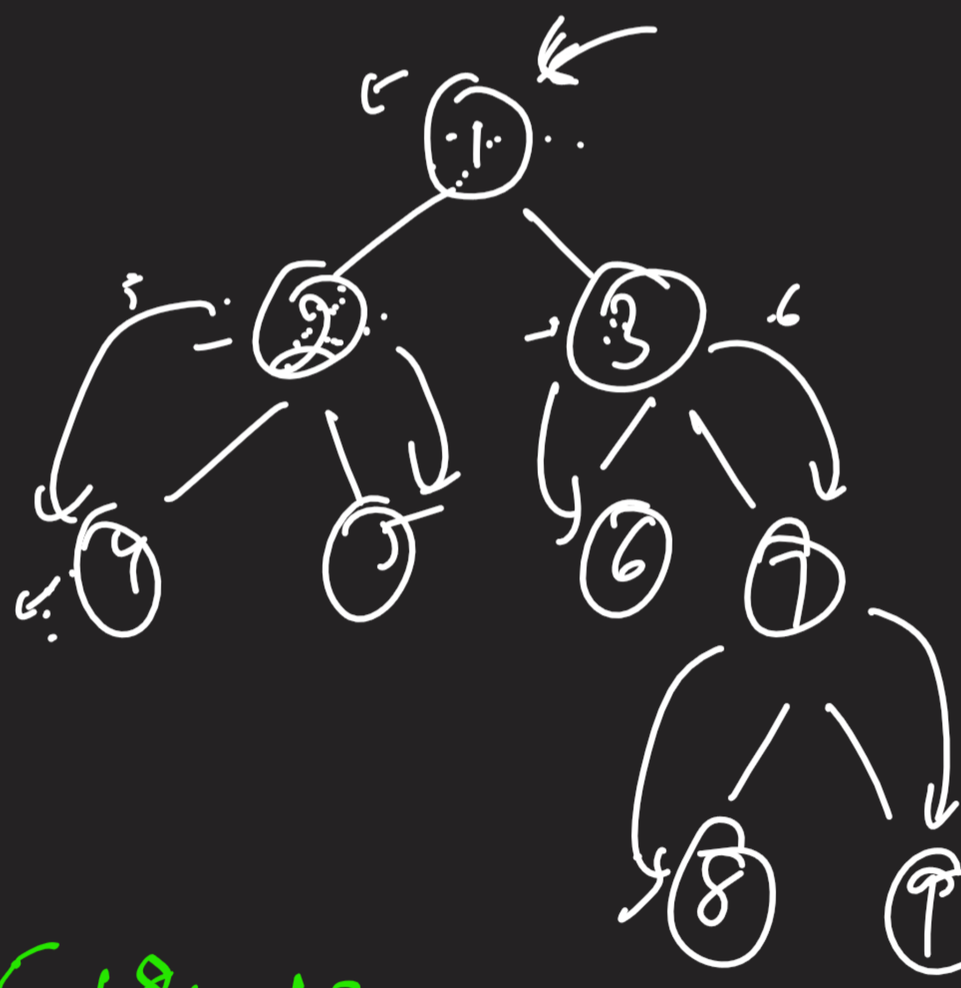
fact(6)
└ 6 x fact(5)
└ 5 x fact(4)
└ 4 x (fact(3))
└ 3 x fact(2)
└ 2 x fact(1)
└ 1 x fact(0)
└ 0 x facf(-1)
└ -1 x fact(-2)

f(5) 5 x 24 = 120
f(4) 4 x 6 = 24
f(3) 3 x fact(2) 2 = 6
f(2) 2 x 1 = 2
f(1) 1 x fact(0)
f(0)

fact(0) = 1

1 x fact(0)

12



fib(0) = 0
fib(1) = 1
fib(2) = fib(1) + fib(0)
fib(3) = fib(2) + fib(1)
fib(n) = fib(n-1) + fib(n-2)

f
f 1
0 1 1 2 3 5 8 13 21
n

0 1 1 2

fib(5) = fib(4) + fib(3) = 5

fib(3) + fib(2)

fib(2) + fib(1)

fib(1) + fib(0)

fib(2) + fib(1)

fib(1) + fib(0)

Time complexity / space complexity

Big (O)

1 + 2 + 3 + 4 + - - - - - + 100000000

T(n) ∝ n

f(n) ∝ n

$T(n) = an + b$

y = mx + c

O(n)

linear time complexity

```
for(int i=0; i<n ; i++){
    for(int j=0; j<n ; j++){
        cout<<"hi";
    }
}
```

n time.

n times of n times of n time

---- n(n times)

i=0 → n times

i=1 → n times

i=2 → n times

i=n-1 → n times

$T(n) = n \times n$ → operation

$O(n^2)$ → quadratic time compl.

$O(\log n)$ →

$O(n \times \log n)$

← $O(1)$ → constant time.

a = 10000

b = 1 →

c = 1+1 → = 2

10000000 + 10 → 3 step

$O(n!)$ → very slow

$O(n^3)$

$O(2^n)$ → very slow.