Q. Given a number N. Find sum of the digits.

Ex N = 45216

4+5+2+1+6 = 18

sumofdigits(45216) = sumofdigits(4521) + 6

sumdig(N) = sumdig($\frac{N}{10}$) + N%10

↳ sum of digits of N

```
int sumdigits(int N)
{
    if(N<10) return N
    OR
    if(N==0) return 0

    return sumdig(N/10) + N%10
}
```

Q. Given a, N. Find a^N. a > 0

a = 2
N = 5

$a^N = 2^5 = 32$

$$a^N = a.a.a \ldots \ldots a$$

(N times)

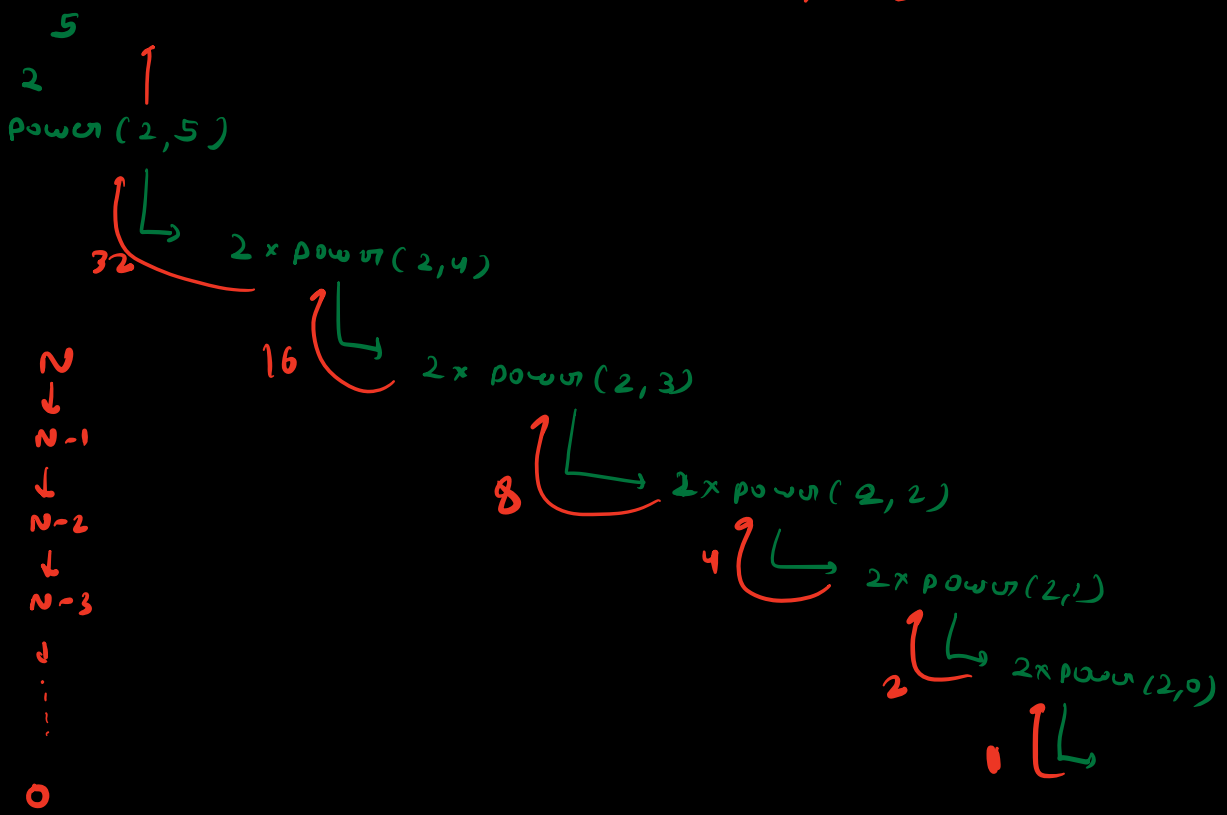
$$a^N = a.a^{N-1}$$

pow(a,N) = a * pow(a,N-1)

a^N

```
int power(a, N)
{
    if(N==0) return 1
    return a * power(a, N-1)
}
```

$2^0 = 1$
$3^0 = 1$
$4^0 = 1$
$a^0 = 1$



TC: no of recursive calls × time taken in each call

$N \times O(1)$

$TC: O(N)$

$$a^{10} = a \times a^{9}$$

$$a^{10} = a^{5}\, a^{5}$$

$$a^{11} = a\, a^{5}\, a^{5}$$

$$a^{101} = a\, a^{50}\, a^{50}$$

$$a^{32} = a^{16}\, a^{16}$$

$$a^{8}\, a^{8}$$

$$a^{4}\, a^{4}$$

$$a^{2}\, a^{2}$$

$$a^{1} \times a^{1}$$

$$a^{0}\, a^{0}$$

$$a^{25} = a\, a^{12}\, a^{12}$$

$$a^{6}\, a^{6}$$

$$a^{3}\, a^{3}$$

$$a\, a^{1}\, a^{1}$$

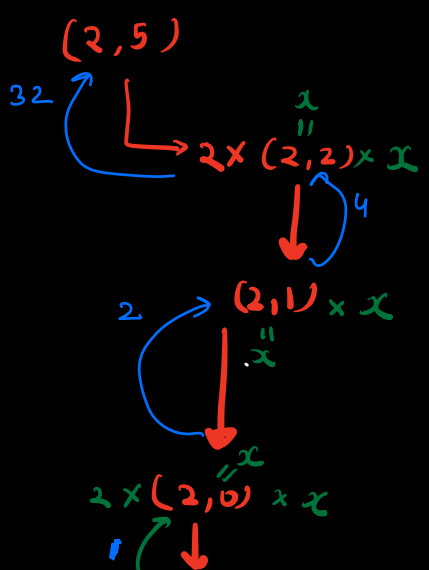
$$a\, a^{0}\, a^{0}$$

$$a^N = \begin{cases} a^{N/2} \cdot a^{N/2} & N \text{ is even} \\ a \cdot a^{N/2} \cdot a^{N/2} & N \text{ is odd} \end{cases}$$

```
int power(a, N)
{
    if (N==0) return 1

    if (N is even)
        return power(a, N/2) * power(a, N/2)
    else
        return a x power(a, N/2) * power(a, N/2)
}
```

```
nt power(a, N)
{
    if (N==0) return 1
    int x = power(a, N/2)
    if (N is even)
        return x * x
    else
        return a * x * x
}
```



TC: $\log N$ $O(1)$

TC: $O(\log N)$

Time complexity.

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

$T(N)$ = time taken to calculate $sum(N)$

$T(N-1)$ = time taken to calculate $sum(N-1)$

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

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

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

$$T(N) = [\,T(N-2) + 1\,] + 1$$

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

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

$$T(N) = T(N-3) + 3$$

$N \rightarrow N-1 \rightarrow N-2 \rightarrow N-3 \rightarrow \cdots \rightarrow 1$

$N \times O(1) = T : O(N)$

After K steps

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

$K = N-1$

$N - K = 1$

$K = N-1$

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

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

$T(N) = N$

TC: $O(N)$

$$T(N) = 2\,T(N-1) + 1$$

$$T(N-1) = 2\,T(N-2) + 1$$

$$T(N) = 2\,[\,2\,T(N-2) + 1\,] + 1$$

$$T(N) = 4\,T(N-2) + 2 + 1$$

$$T(N) = 4\,T(N-2) + 3$$

$$T(N-2) = 2\,T(N-3) + 1$$

$$T(N) = 8\,T(N-3) + 7$$

$T(0) = 1$

$$T(N) = 2^{K}\,T(N-K) + (2^{K} - 1)$$

$k = N$

$N - k = 0$

$k = N$

$$T(N) = 2^N \, T(0) + 2^N - 1$$

$$= 2^N + 2^N - 1$$

$$T.C: O(2^N)$$

$$T(N) = T\left(\frac{N}{2}\right) + 1$$

$T(1) = 1$

$$T\left(\frac{N}{2}\right) = T\left(\frac{N}{4}\right) + 1$$

$$T(N) = T\left(\frac{N}{4}\right) + 2$$

$$T\left(\frac{N}{4}\right) = T\left(\frac{N}{8}\right) + 1$$

$$T(N) = T\left(\frac{N}{8}\right) + 3$$

Break
10:16 10:26

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

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

$K = \log N$

$$N = 2^k$$

$$2^k = N$$

$$k = \log N$$

$$T(N) = T(1) + \log N$$

$$T(N) = 1 + \log N$$

$$T(N) = O(\log N)$$

$$T(N) = 2T\left(\frac{N}{2}\right) + 1$$

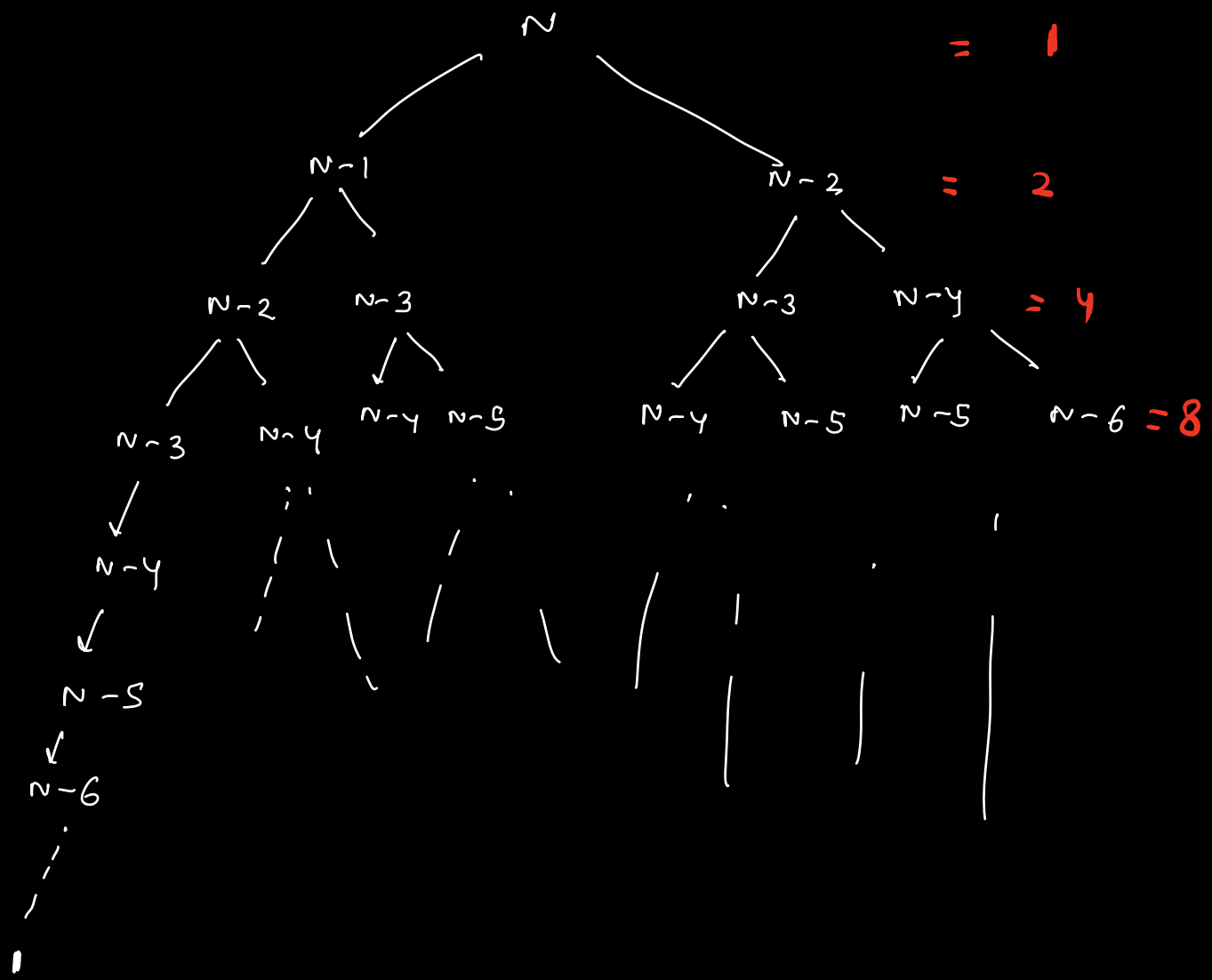
$$T(N) = T(N-1) + T(N-2) + 1$$

$$T(N) = 2\,T(N-1) + 1$$

TC: $O(2^N)$

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

No of recursion calls $\times$ $O(1) = 2^N$



N levels.

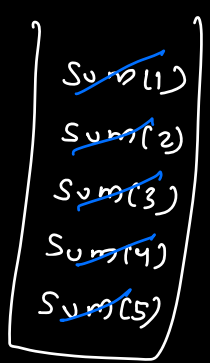
$1 + 2 + 4 + 8 + \dots\dots 2^N$

$$S_N = \frac{a(r^N - 1)}{r - 1} = \frac{1(2^N - 1)}{2 - 1} = 2^N - 1$$

TC : $O(2^N)$

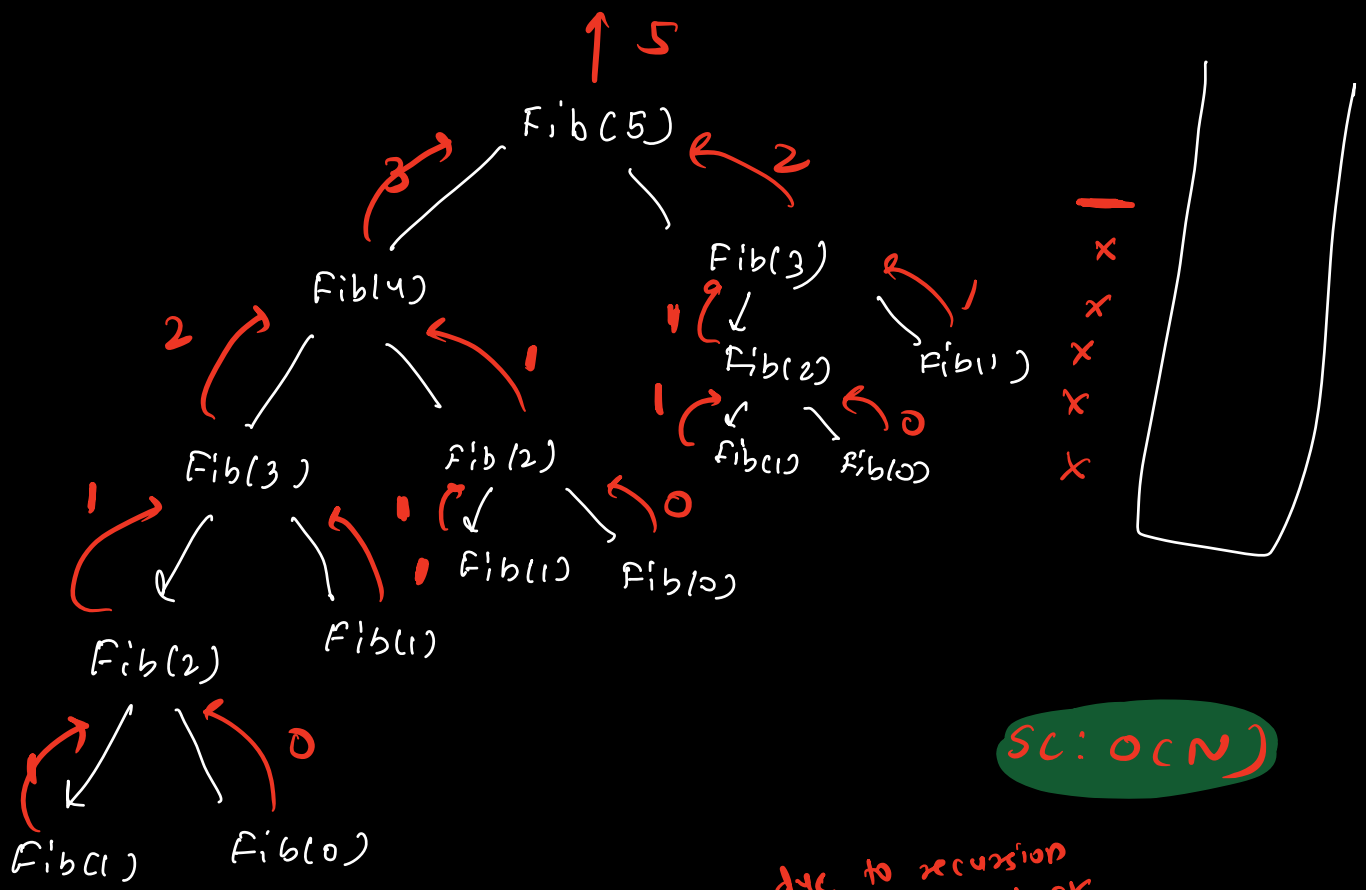
Space complexity. [function stack , consider both due to each call]

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



SC: O(N)

SC = No of function calls



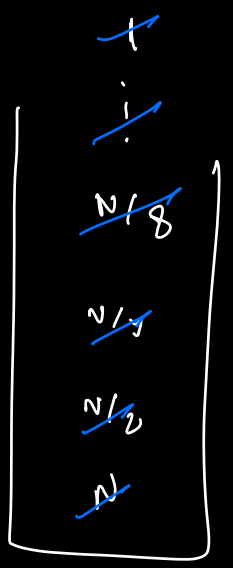
SC: O(N)

due to recursion stack

SC: max stack size at any point

or

No of level recursive tree.

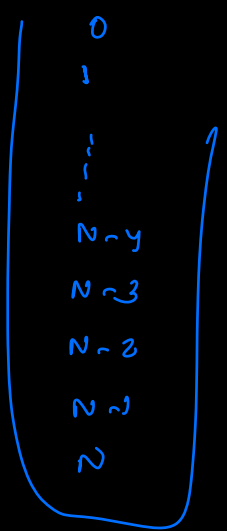


SC: O(log N)

Doubts

```
int factors ( )
{
    int arr[N]
}

int Fib(n)
{
    int x = factors(n)
    return fib(N-1) + fib(N-2)
}
```

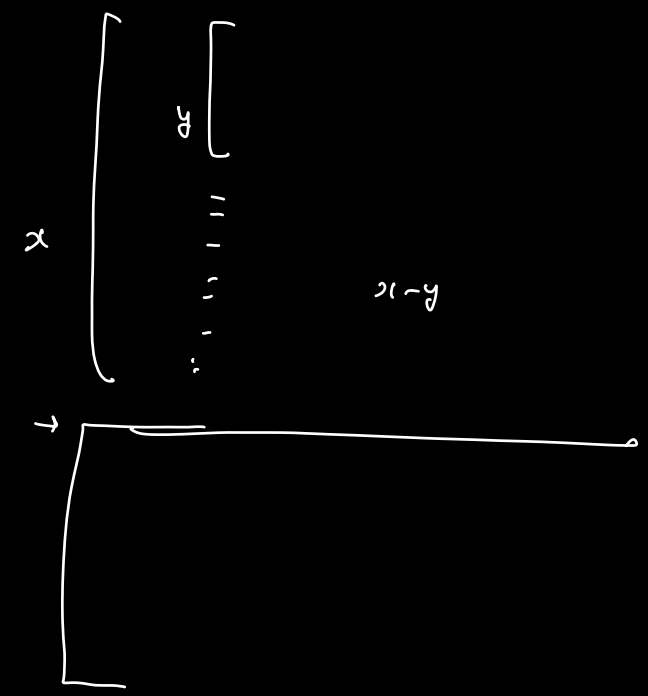


SC: N + N = O(N)

$2^N \neq N$

$N \times N$

$TC: O(1)$



23rd dec – 3rd Jan