

Recursion-2



How to Think Recursively - - > Transforming into Recursive Code

Base case is very important

① Time Complexity :- of recursion program is directly proportional to the number of functions calls that are made before terminates

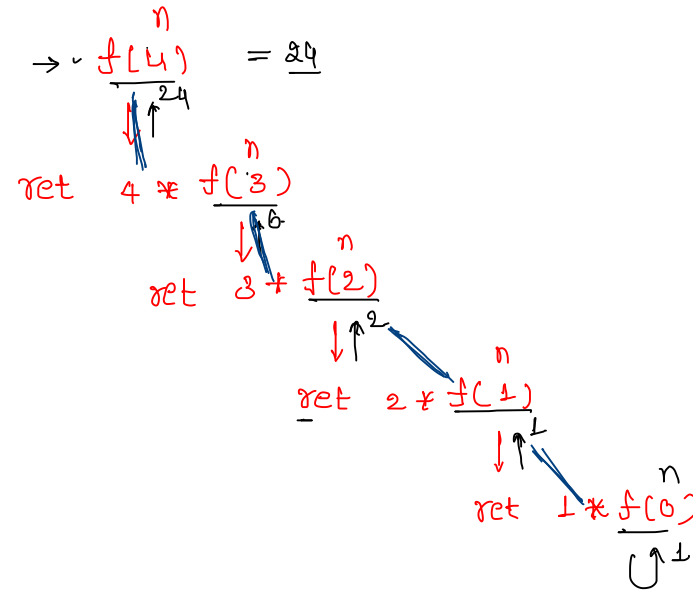
② Space Complexity :- of recursion program is directly proportional to the height of stack

$$\begin{array}{l} O(n^2) \left\{ \begin{array}{l} \approx 25 \\ \approx 100 \end{array} \right. \leftarrow \begin{array}{l} n=5 \\ n=10 \end{array} \rightarrow \begin{array}{l} 5 \text{ fn calls} \\ 10/11 \end{array} \left. \vphantom{\begin{array}{l} n=5 \\ n=10 \end{array}} \right\} O(n) \end{array}$$

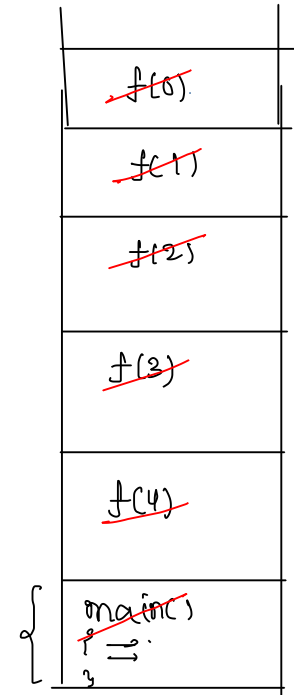
$$\begin{array}{l} \vdots \\ \text{approximately} \\ \begin{array}{l} n=4 \rightarrow 16 \text{ fn. } (2^4) \\ n=5 \rightarrow 32 \text{ fn } (2^5) \\ n=10 \rightarrow 1024 \text{ fn calls } (2^{10}) \end{array} \left. \vphantom{\begin{array}{l} n=4 \\ n=5 \\ n=10 \end{array}} \right\} \frac{O(2^n)}{\text{exponential T.C}} \end{array}$$

1) Factorial program

```
int fact(int n)
{
    if (n==0) // stop x ✓
    {
        return 1;
    }
    else
    {
        return n*fact(n-1);
    }
}
```



Approximately n fn. calls
 $\therefore O(n)$ [T.C]



Approximately n activation records

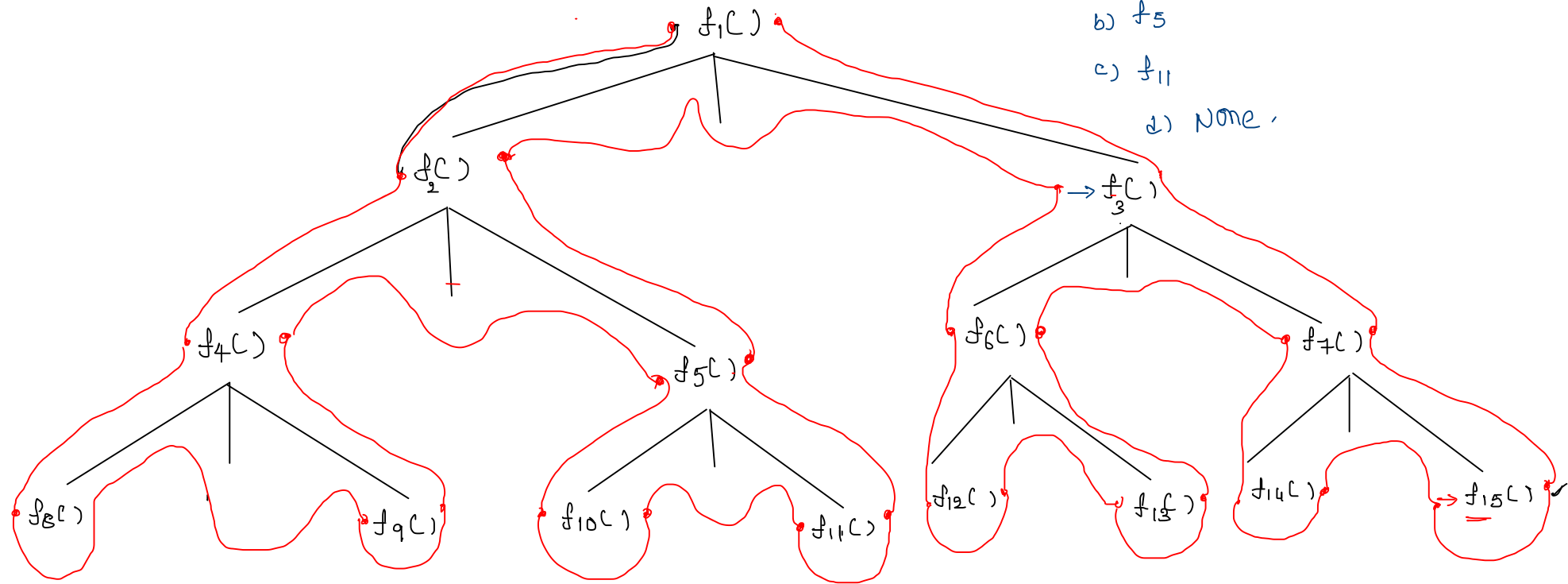
so s.c: $O(n)$

max
depth of
rec. tree

for every function call, there will be one activation record will be push into the stack

for some program. (let's not worry)

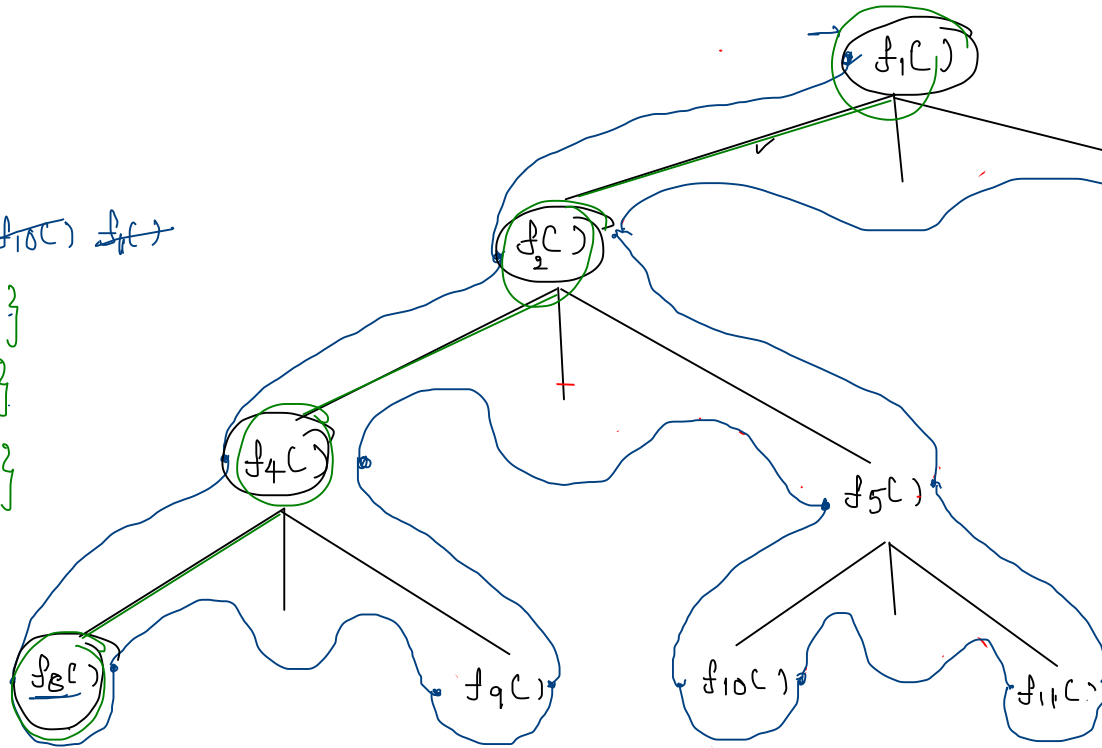
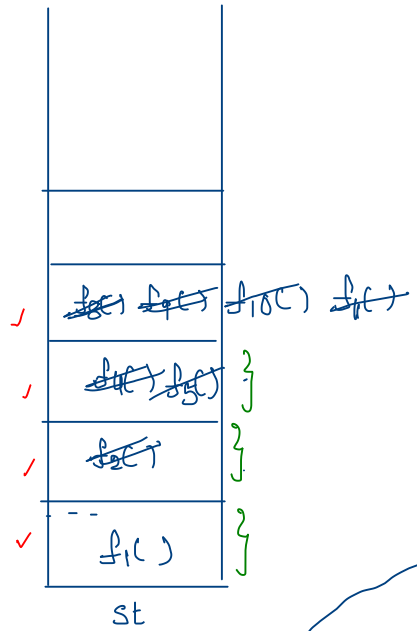
- ✓ a) f_2 ✓
- b) f_5
- c) f_{11}
- d) None.



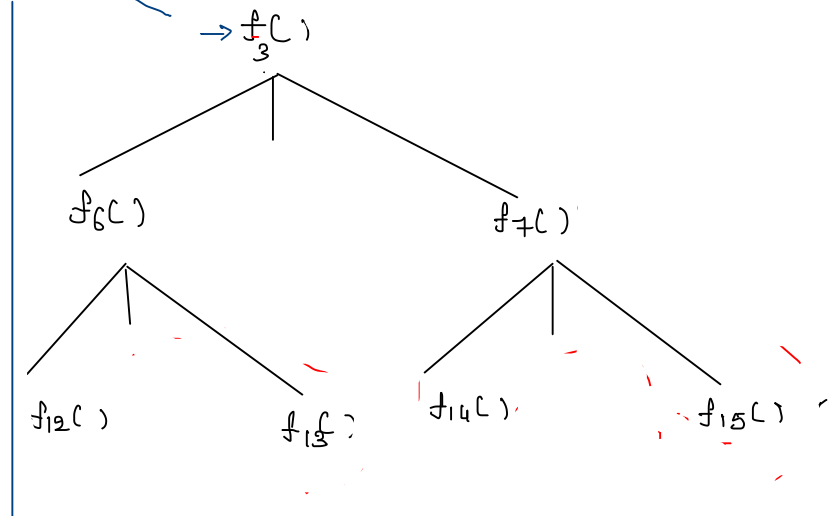
recursion tree

top to down
Left to Right

for some program. (let's not worry)



- ✓ a) f₂ ✓
- b) f₅
- c) f₁₁
- d) None



3+1 = 4 cells
 max (Height) of stack
 (or) we used
 max depth
 of recursion tree

recursion tree

top to down
 Left to Right

2) Implement pow(a,b) function [a>=1 and b>=0] :-

$$\text{pow}(2,3) = 8 \quad \hookrightarrow 2 \times 2 \times 2$$

$$\text{pow}(3,0) = 3 \quad \hookrightarrow 3^0 = 1$$

$$\text{pow}(2,5) = 2^5 = 32$$

$$= \underbrace{2}_{a} \times \underbrace{2 \times 2 \times 2 \times 2}_{\substack{\hookrightarrow \text{pow}(2,4) \\ a \quad b-1}}$$

$$\text{pow}(2,5) = 2 * \text{pow}(2,4)$$

$$\text{pow}(2,4) = 2 * \text{pow}(2,3)$$

⋮

$$\text{pow}(2,0) = 1$$

$$\text{pow}(3,4) = 81$$

```
int pow(int a, int b) // a power b
{
```

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

```
    else
        return a*pow(a,b-1);
```


```
}
```

```
int pow(int a, int b)
{
    if(b==0)
        return 1;

    else
    {
        return a*pow(a,b-1);
    }
}
```


3) Find the sum of all digits of given number

n=12345 == > 1+2+3+4+5 = 15 [output]

```
int sum(int n)
{
     sum()
}
```

```
}
```

Greatest Common Divisor

(or)

HCF

studied ✓

4) Implement GCD(a,b) :-

$$\text{GCD}(24, 36) = ?$$

int GCD(int a, int b)

{

4th 5th → Recursion

prime factorization

$$\begin{aligned} 24 &= \underline{2} \times \underline{2} \times \underline{2} \times \underline{3} \\ 36 &= \underline{2} \times \underline{2} \times \underline{3} \times \underline{3} \end{aligned} \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \begin{array}{l} 2 \times 2 \times 3 \\ = 12 \checkmark \end{array}$$

$$2 \times 2 \times$$

$$44 =$$

$$48$$

$$\begin{array}{r} 24 \overline{) 36} (1 \\ \underline{- 24} \\ 12 \end{array} \quad \begin{array}{r} 12 \overline{) 24} (2 \\ \underline{- 24} \\ 0 \end{array}$$

↓

$$\begin{array}{r} 32 \overline{) 56} (1 \\ \underline{- 32} \\ 24 \end{array} \quad \begin{array}{r} 24 \overline{) 32} (1 \\ \underline{- 24} \\ 8 \end{array}$$

$$\begin{array}{r} 48 \overline{) 56} (1 \\ \underline{- 48} \\ 8 \end{array}$$

↓

$$\text{GCD}(\underline{12}, \underline{17}) = 1$$

$$\frac{17}{12} \text{ } | \text{ } r=5$$

$$\begin{array}{r} 17 \\ 12 \overline{) 17} \\ \underline{-12} \\ 5 \end{array}$$

remainder \rightarrow $\textcircled{5}$

$b \% a$

$$\begin{array}{r} 12 \\ 5 \overline{) 12} \\ \underline{-10} \\ 2 \end{array}$$

$$\begin{array}{r} 5 \\ 2 \overline{) 5} \\ \underline{-4} \\ 1 \end{array}$$

$$\begin{array}{r} 2 \\ 1 \overline{) 2} \\ \underline{-2} \\ 0 \end{array}$$

$$0 \text{) } \textcircled{1} \text{ }$$

stop

$$\text{GCD}(\underline{12}, \underline{17})$$

$$\text{GCD}(\underline{5}, \underline{12})$$

$$\text{GCD}(\underline{2}, \underline{5})$$

$$\text{GCD}(\underline{1}, \underline{2})$$

$$\text{GCD}(\underline{0}, \underline{1})$$

$$\begin{array}{c} a \quad b \\ \text{GCD}(a, b) \\ \downarrow \\ \text{ret. GCD}(a, b \% a) \end{array}$$

$$\text{GCD}(b-a, a)$$

$$\text{GCD}(\underline{17}, \underline{12})$$

$$\text{ret GCD}(\underline{12}, \underline{17})$$

$$\underline{12} \% \underline{17} = 12$$

normal flow

```
int GCD(int a, int b)
{
    if(a==0)
        return b;
    return GCD(b%a, a);
}
```

Euclid
Mathematician
Algo

any two consecutive fibonacci numbers are w.c i/p to this Algo

GATE CSE 2021 Set 2 | Question: 23

asked in Algorithms Feb 18, 2021 • retagged Nov 30, 2022 by Lakshman Bhaiya



9



Consider the following ANSI C function:

```
int SomeFunction (int x, int y)
{
    if ((x==1) || (y==1)) return 1;
    if (x==y) return x;
    if (x > y) return SomeFunction(x-y, y);
    if (y > x) return SomeFunction (x, y-x);
}
```

The value returned by SomeFunction(15, 255) is _____

repetitive subtraction

GCD ✓

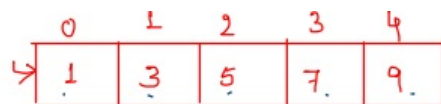
$$\begin{array}{r} x \quad y \\ S(24, 36) \\ \hline 12 \\ \downarrow \\ S(24, 12) \\ \hline 12 \\ \downarrow \\ S(12, 12) \\ \hline \downarrow \uparrow 12 \end{array} = 12$$

Array based recursion problems

5) Find sum of all elements of array

```
int sum(int arr[], int n)
{
    if(n==1) return arr[0];

    return arr[n-1]+sum(arr,n-1);
}
```

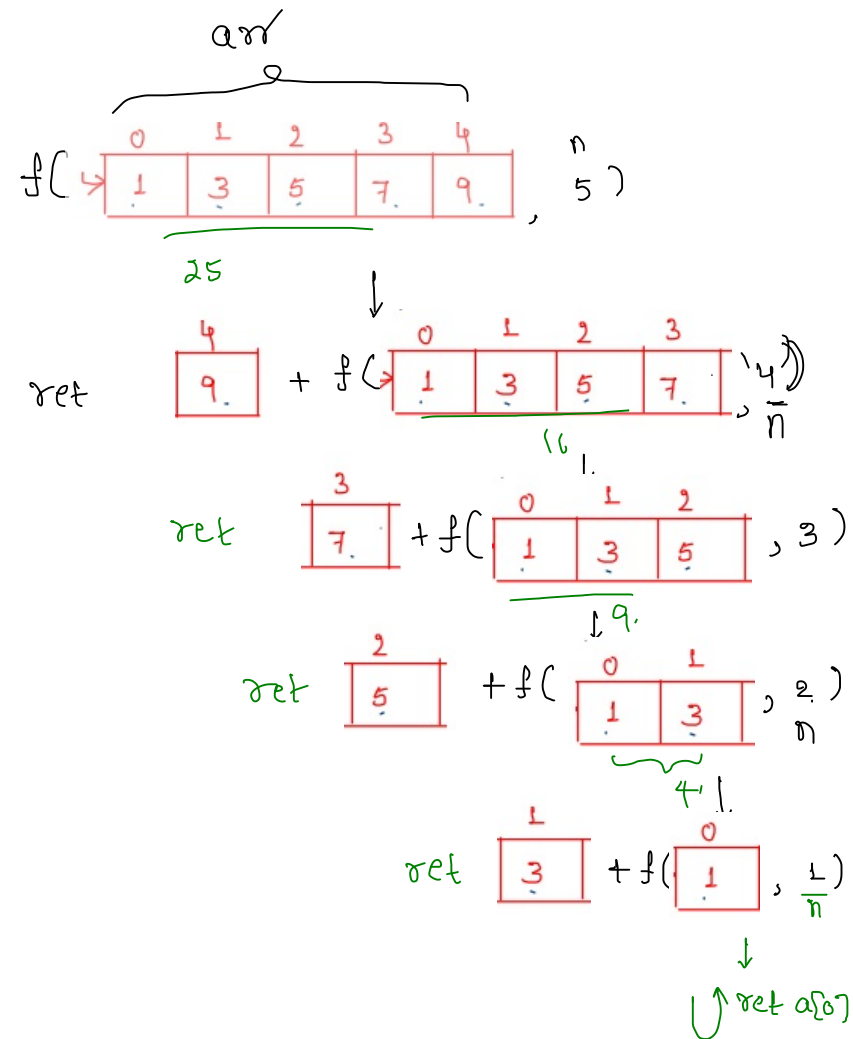


$s(a, n)$ \uparrow # of eles

$\{ 9 + s(a, 4)$

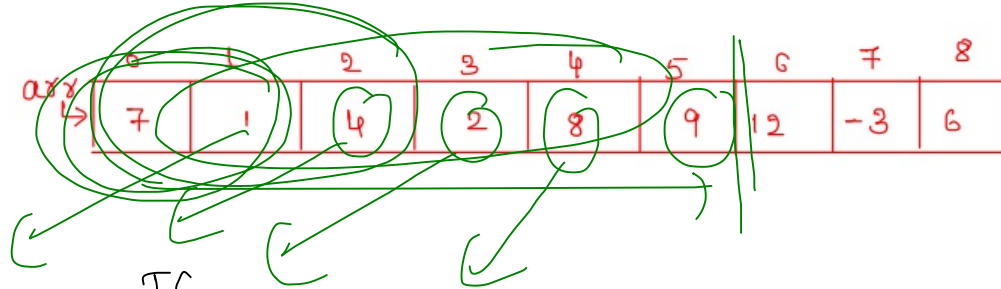


$a[n-1] + s(a, n-1)$



6) Find maximum element of an array using recursion

$n=9$



```

int findMax(int arr[], int n)
{
    int res=arr[0];
    for(int i=1;i<n;i++)
    {
        res=Math.max(arr[i],res);
    }
    return res;
}

```

TC

$\Rightarrow O(n)$

SC : $O(1)$

```

int findMax(int arr[], int n)
{
    if(n==1)

```

```

        return arr[0];
    else

```

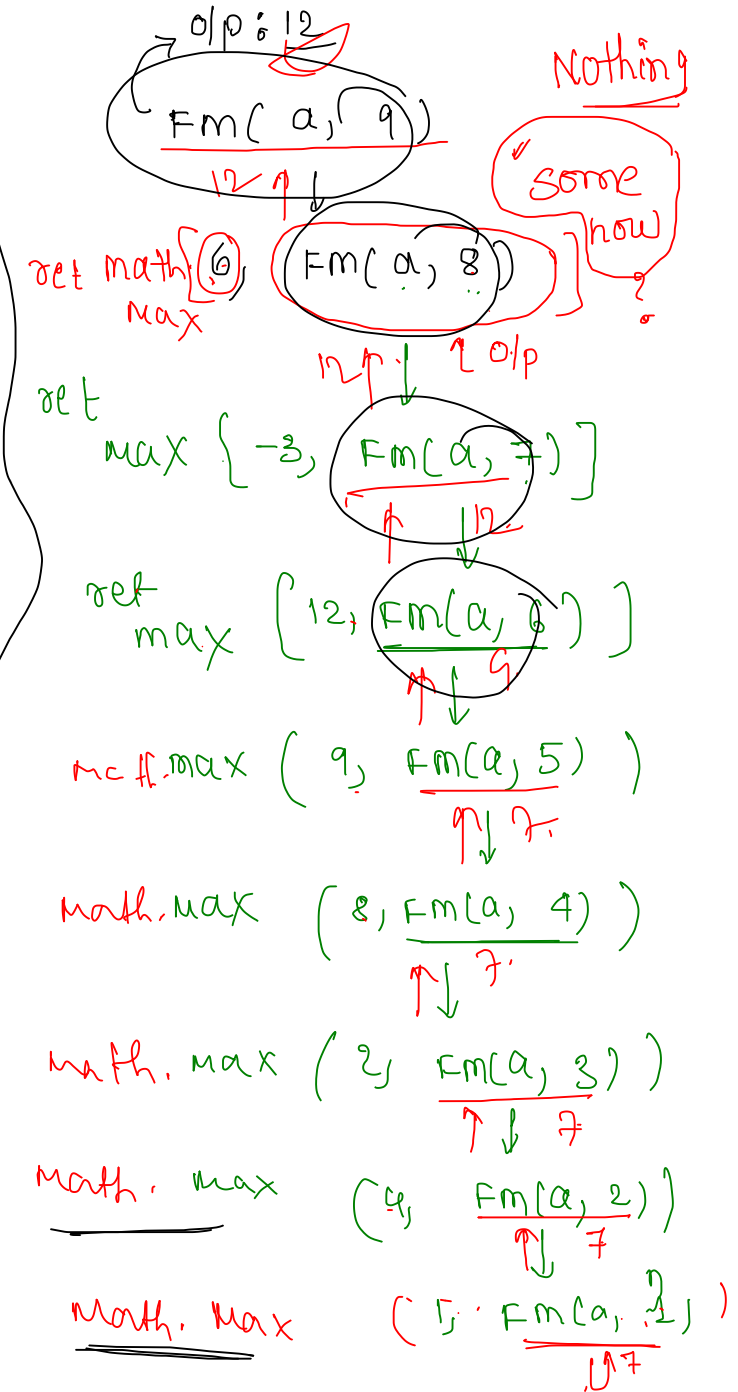
```

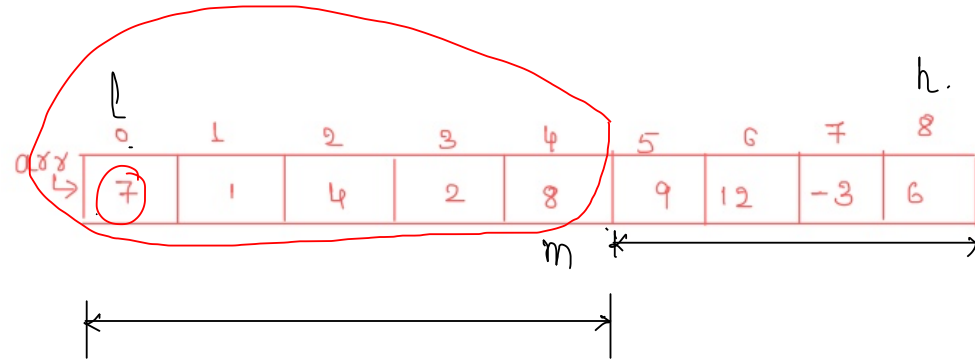
        return Math.max(arr[n-1], findMax(arr,n-1));
    }
}

```

$\Rightarrow O(n)$ TC

SC : $O(n)$ [Recursive stack space]





$$\frac{0+8}{2} = 4$$

~~some
now~~

$$fm(a, 0, 8) = 12$$

$m=4$

$$t_1 = fm(a, 0, 4) \quad t_2 = fm(a, 5, 8) \quad \text{ret } \max(t_1, t_2)$$

$\uparrow 8$ $\uparrow 12$

$$\frac{l+h}{2} = \frac{0+1}{2} < 0$$

$m=2$

$$t_1 = fm(a, 0, 2)$$

$$t_2 = fm(a, 3, 4)$$

$m=1$

$$t_1 = fm(a, 0, 1)$$

$$t_2 = fm(a, 2, 2)$$

$m=0$

$$t_1 = fm(a, 0, 0)$$

$$t_2 = fm(a, 1, 1)$$

✓ $\text{return Math.max}(\underbrace{FM(arr, low, mid)}_{t_1}, \underbrace{FM(arr, mid+1, high)}_{t_2});$

```
public int fun(int arr[], int low, int high)
{
    if(low==high)
        return arr[low];
    if(low<high)
    {
        int mid=low+(high-low)/2;
        return Math.max(fun(arr,low,mid), fun(arr,mid+1,high));
    }
    return Integer.MIN_VALUE;
}
```