CLASS 44

28 March 2023 21:06

Class - 44

left = 5 Right = 7

5 & 6 & 7

5 → 1 0 1

6 → 1 1 0

7 → 1 1 1

1 0 0 → 4

Approach:

Brute Force

```
int ans = left
For ( i=left+1; i<=right; i++)
{
    ans = ans & i;
}
return ans;
```

10^9 X

T.C =

O(right - left)

Optimized Approach

1) If any bit in the range [L-R] is 0 then that bit in final ans is also 0.

	3	2	1	0
0 →	0	0	0	0
1 →	0	0	0	1
2 →	0	0	1	0
3 →	0	0	1	1
4 →	0	1	0	0
5 →	0	1	0	1
6 →	0	1	1	0

7 →	0	1	1	1
8 →	1	0	0	0
9 →	1	0	0	1
10 →	1	0	1	0

1 & 1 & 1 & 1 & 1 = 1

1 & 0 & 0 & 1 = 0

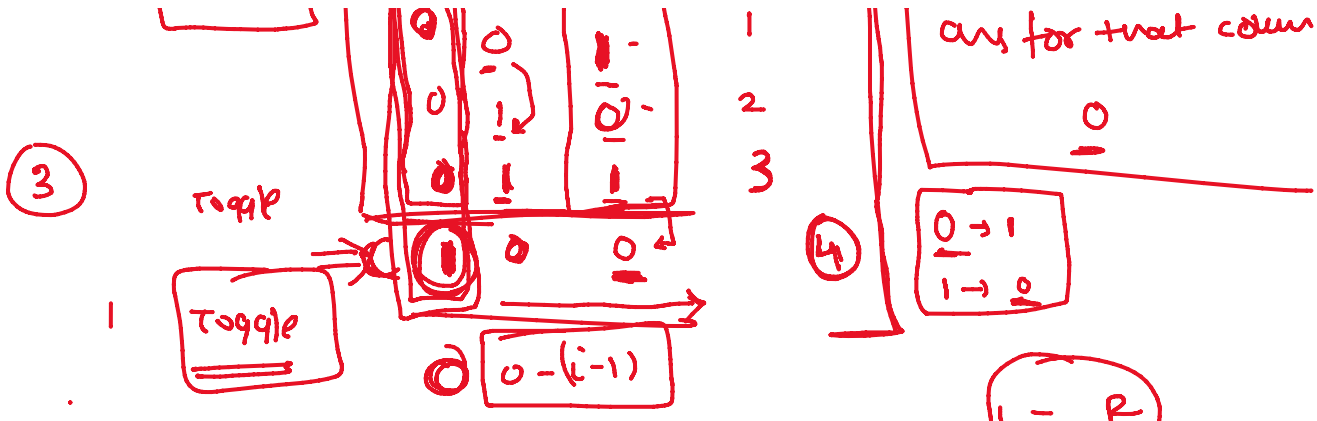
2)

i/p

- -

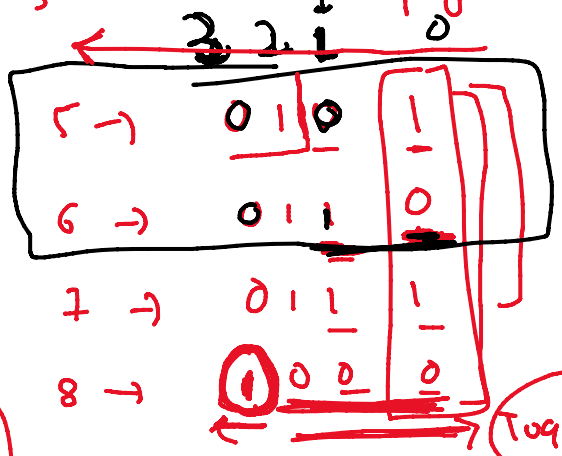
0	0	0	0
0	0	1	1

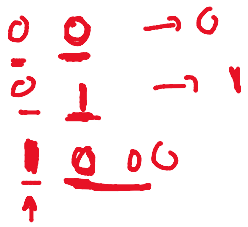
if a bit gets toggled ans for that coun is



if i^{th} column is toggled from 0 - (i-1)

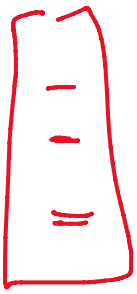
colums already got toggled





i^{th} colum toggling

0 - (i-1)

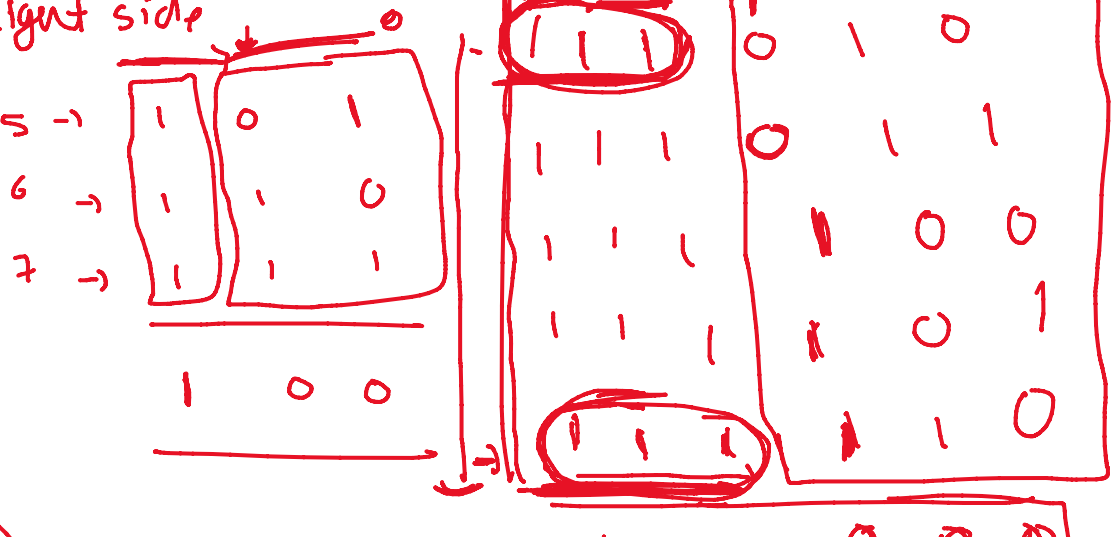


i^{th} colum ⇒ 0 - (i-1) toggling

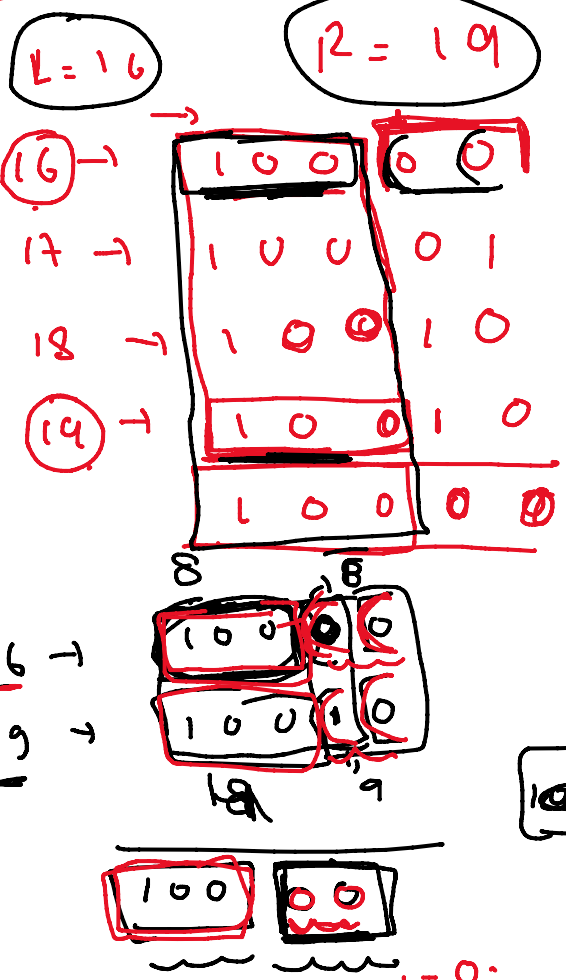
11
0

9 →

Right side



L - R



1 1 1 0 0 0 0 8

16 and 19

2bit

100 << 2 → 100 00

100 → 32

32

O(1)

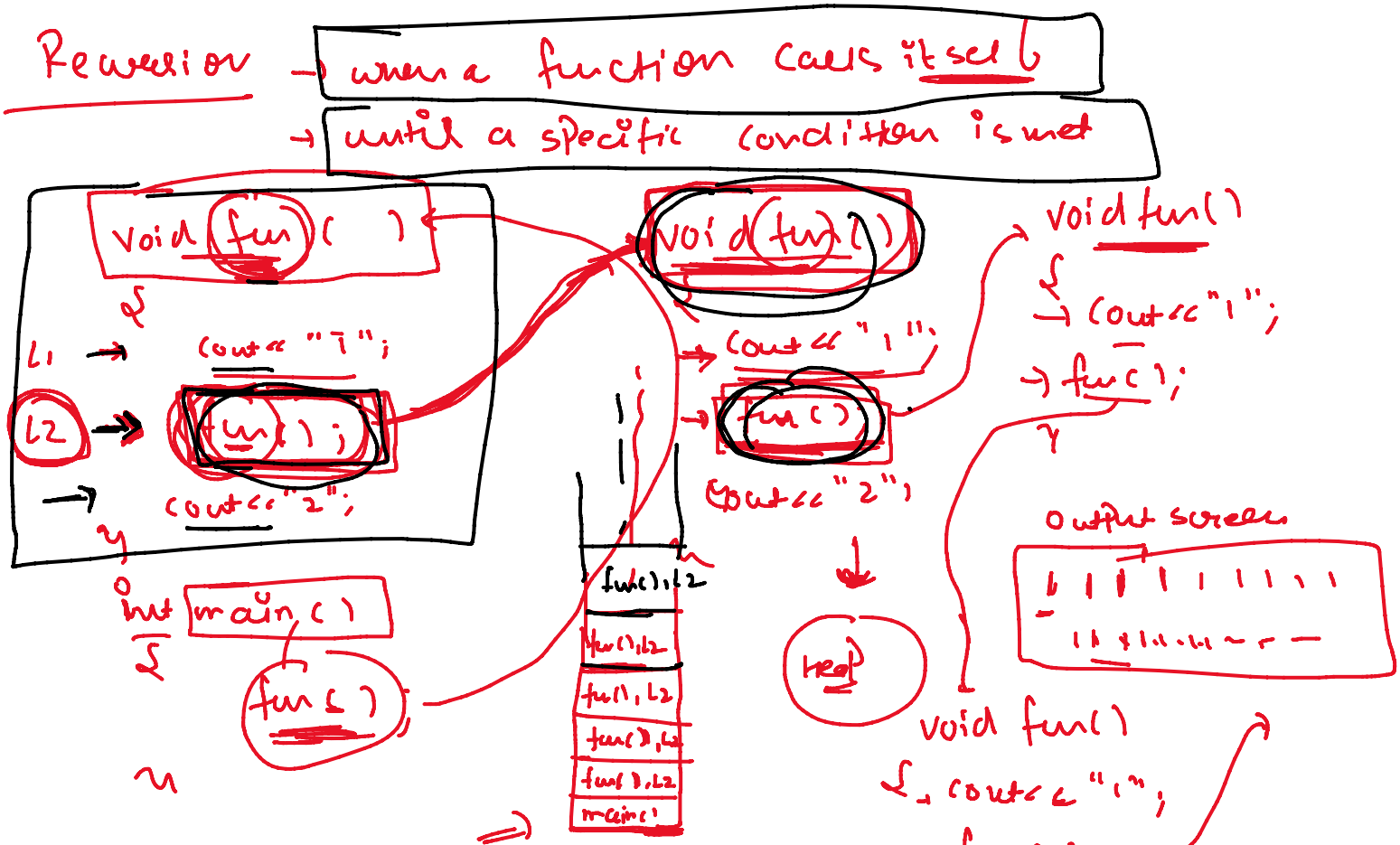
```
int count = 0;
while (m != n)
{   m >>= 1;
    n >>= 1;
    count++;
}
int ans = m << count;
return ans;
```

Recursion

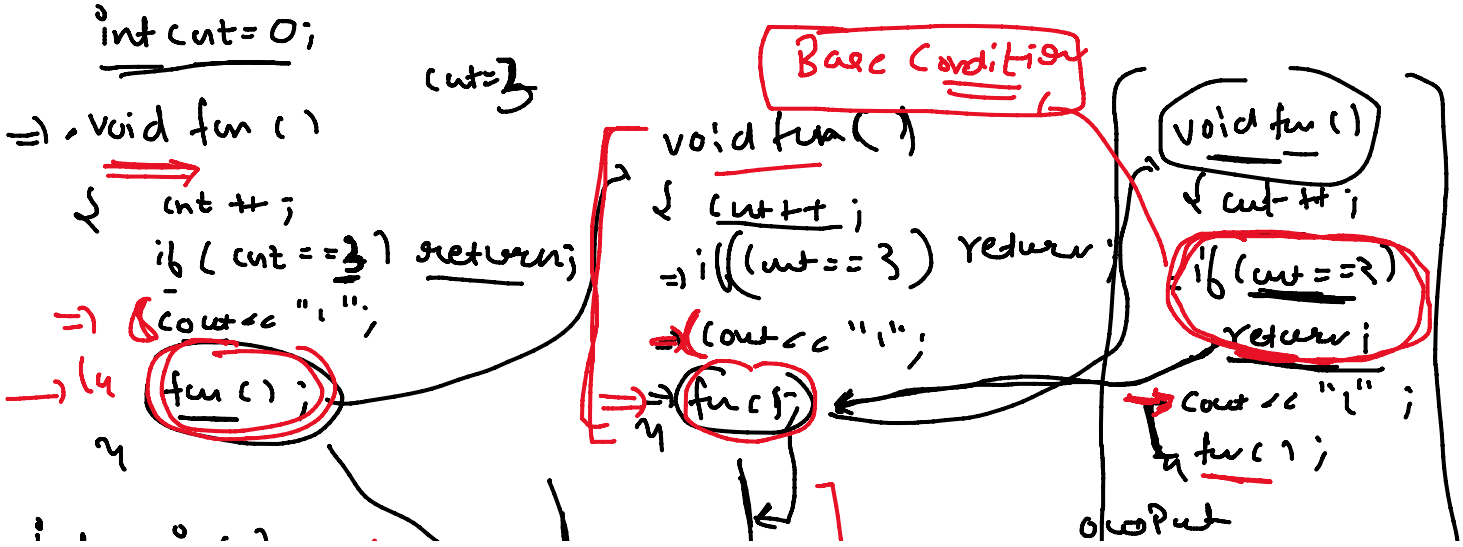
function (a,b)

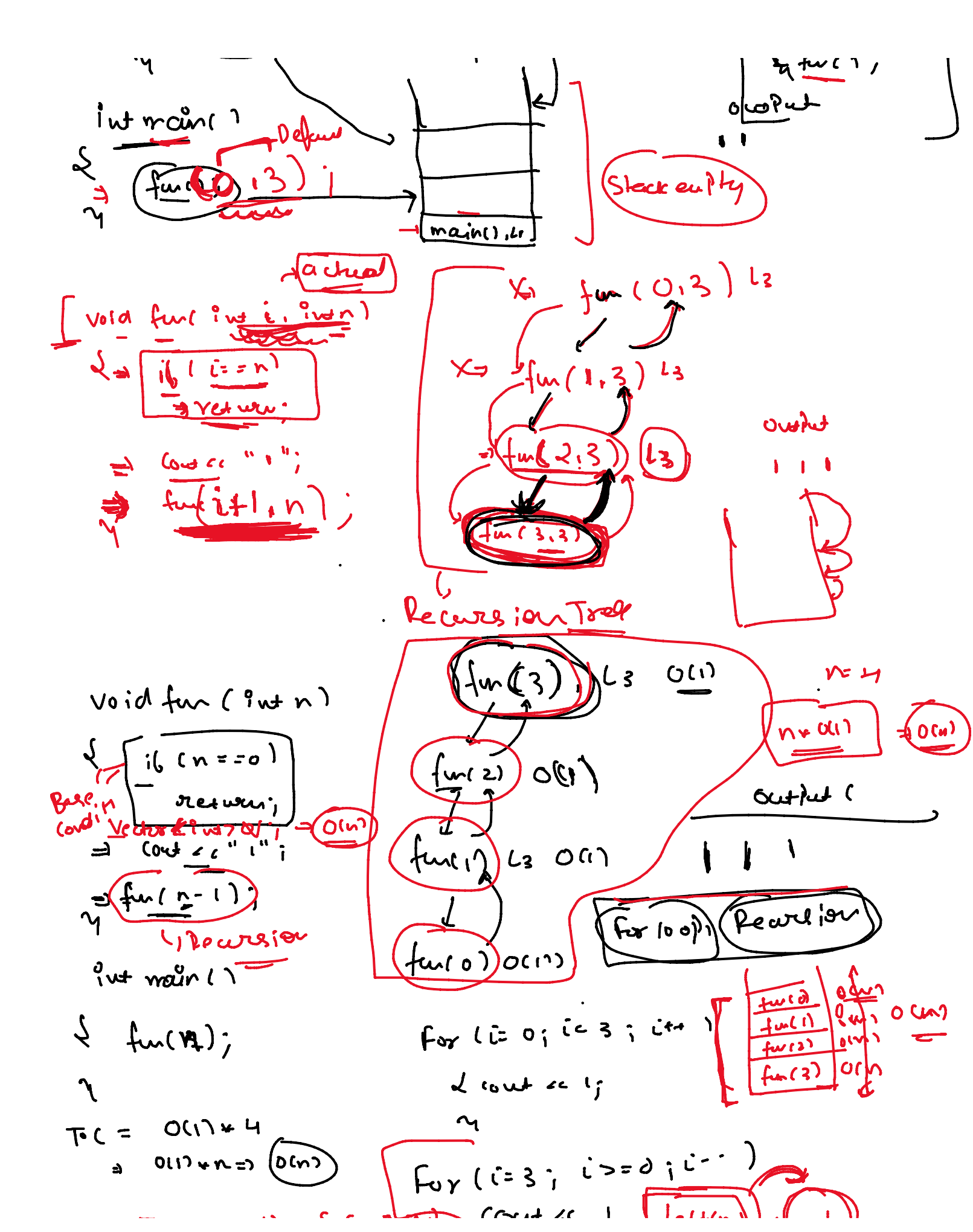
fun (a,b--- ( )

Recursion → when a function calls itself
→ until a specific condition is met



Stack overflow
segmentation fault




int main()
Defined
fun(0,3);
main(),L1
Stack empty
output
actual
void fun(int i, int n)
if(i==n)
return;
cout<<" i ";
fun(i+1,n);
fun(0,3) L3
fun(1,3) L3
fun(2,3) L3
fun(3,3)
output
Recursion Tree
void fun(int n)
if(n==0)
return;
Base condition
cout<<" i "<<i;
fun(n-1);
Recursion
O(n)
fun(3) L3 O(1)
fun(2) O(1)
fun(1) L3 O(1)
fun(0) O(1)
n=4
n*O(1)
O(n)
output
For loop
Recursion
int main()
fun(n);
TC = O(1)*4
O(1)*n => O(n)
for(i=0; i<3; i++)
cout<<i;
fun(0)
fun(1)
fun(2)
fun(3)
O(n)
for(i=3; i>=0; i--)

a O(1) * n => O(n)

T.C = O(N) S.C = O(N)

```
For ( i = 3 ; i >= 0 ; i-
    Cout << 1
```

fact(n) => n!

fact(0) => 1

1!

Find Factorial of NO

5! = 5 * 4 * 3 * 2 * 1

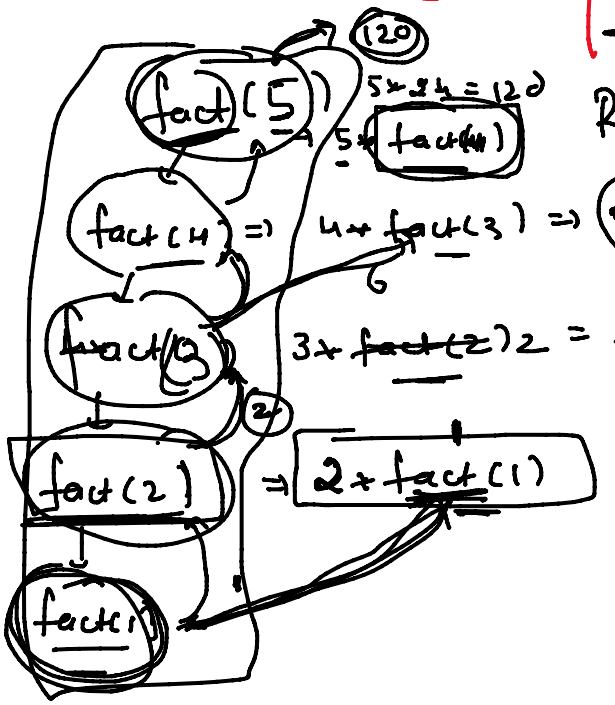
```
int ans = 1
For ( i = 5 ; i >= 1 ; i-- )
{
    ans = ans * i ;
}
```

Recursion

```
int fact ( int n )
{
    if ( n == 1 )
        return 1 ;
    intans = n * fact(n-1) ;
}
```

n! = n * (n-1)!

(n-1)!



Recursive Relation

```
void
int fact (int n)
{
    if(n == 1)
        return 1;
    return n * fact(n-1)
}
```

1

2 * 1 = 2