Recursion

- What is Recursion
- Example of recursion
- Tracing recursion
- Stack used in Recursion
- Time complexity
- Recurrence Relation

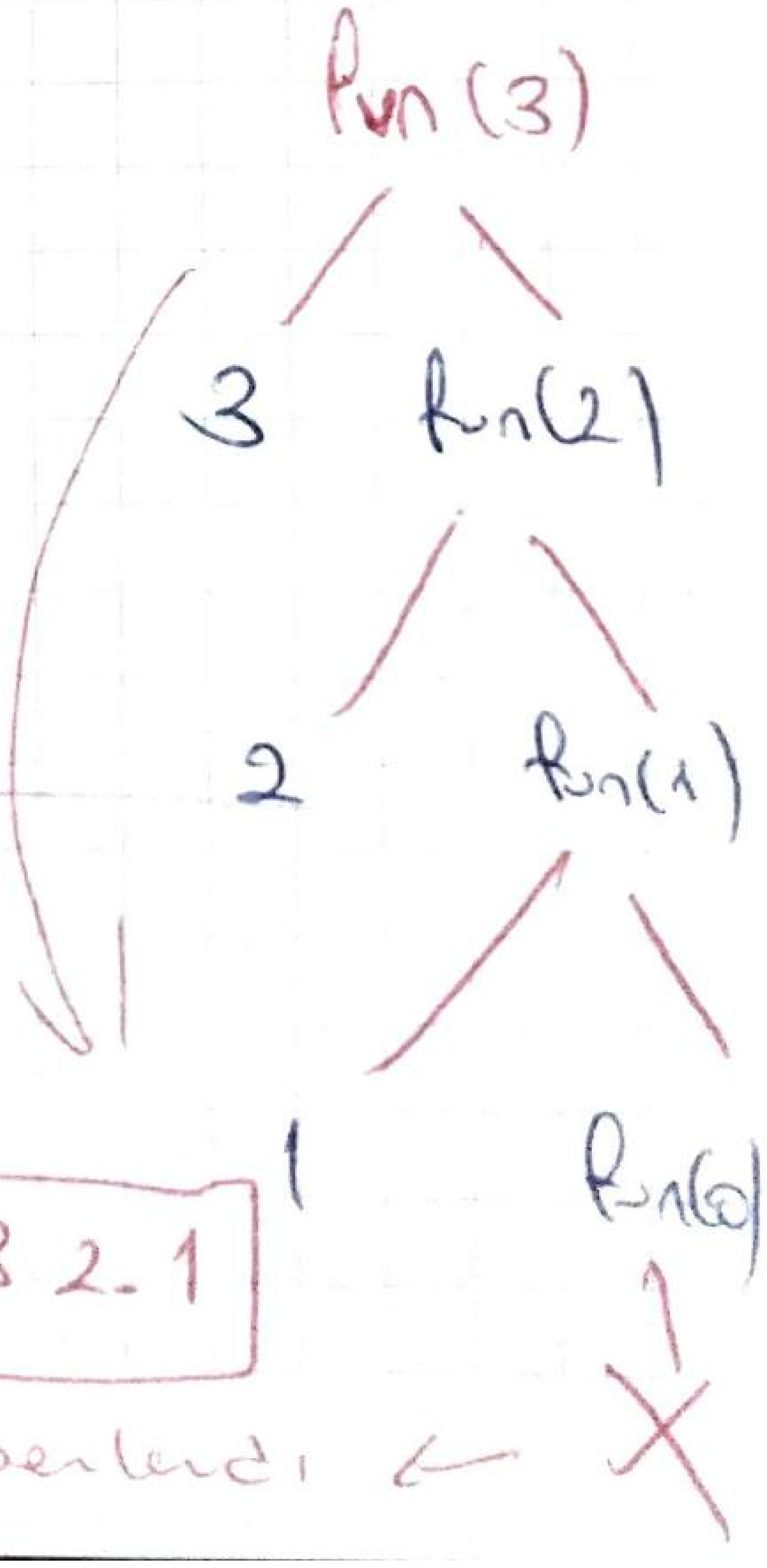
```
type   fun (param) {
          if (<base condition>) {
               ~~1.~~ ___
               ~~2.~~ fun( param);
               3. ___
          }
     }
```

Bu koşul recursiondan çıka bilmesi için

①

```
void   fun ( int  n ) {
          if(n>0){
          1.    cout<<n<<endl;
          2.    fun (n-1);
          }
       }

void   main () {
     int  x=3;
     fun(3);
  };
```

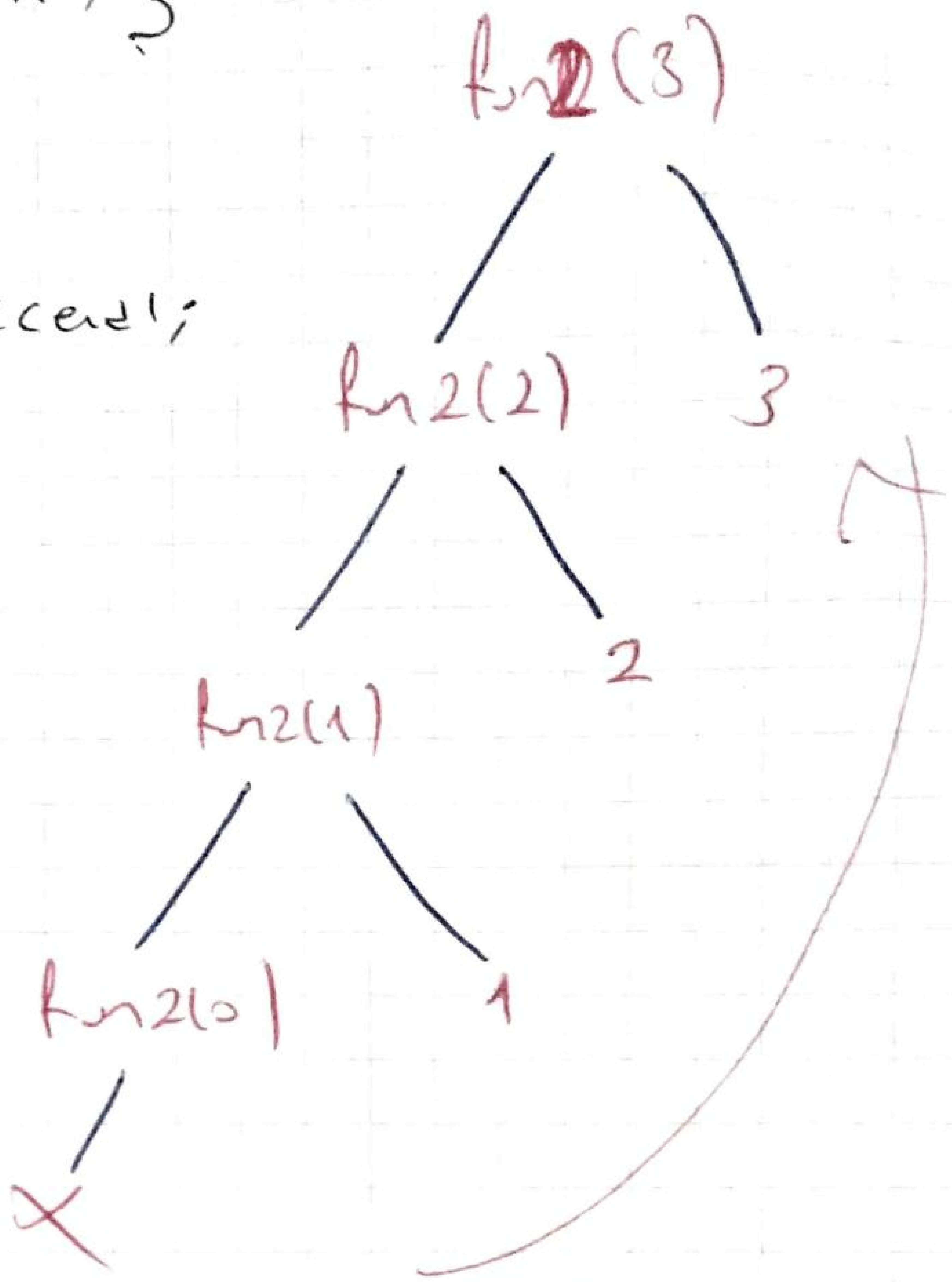


O/p = 3 2.1

2

```
void fun(int n) {
        if(n>0) {
            fun(n-1);
            cout<<n<<endl;
        }
    }
```

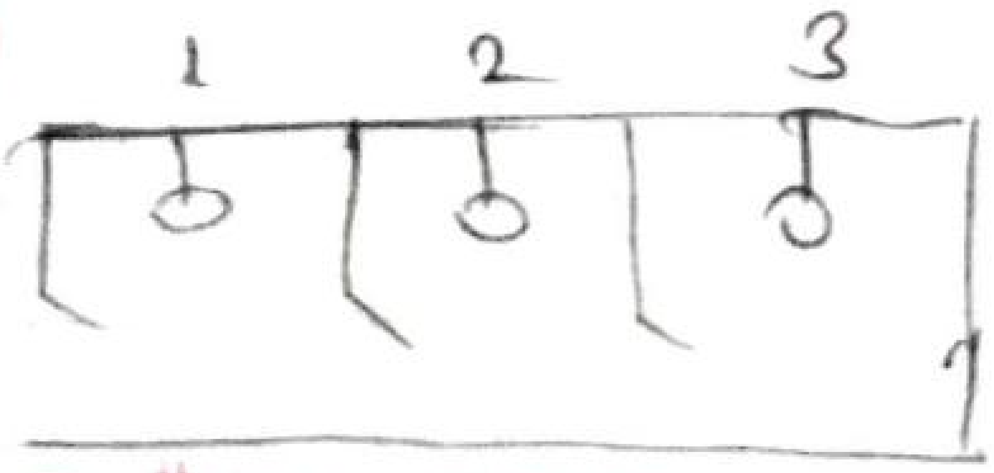
```
void    main() {
     int  x=3
       fun(x)
   };
```



O/p = 1.2.3

basit bir örnek

①



Calling phase

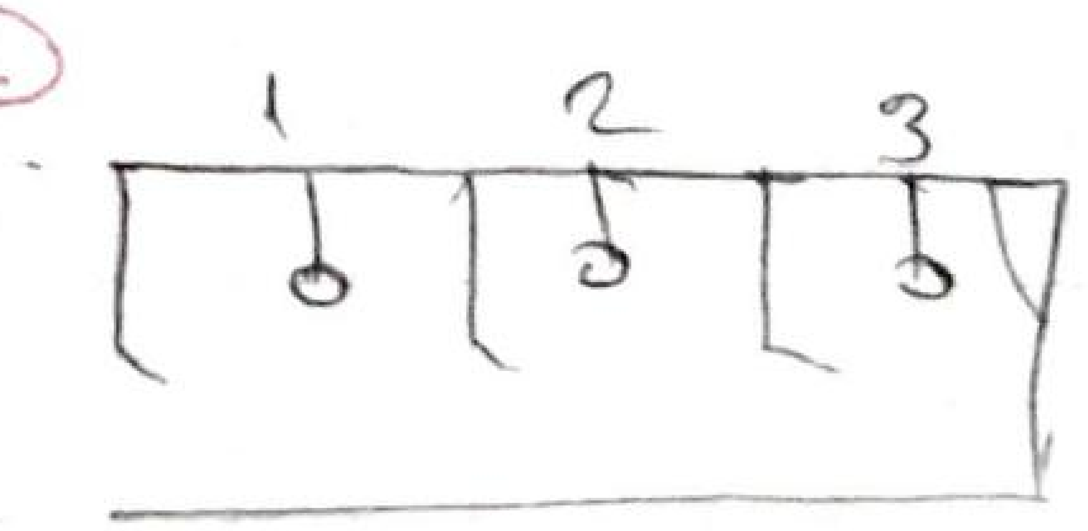
1- lambayı aç

2- ~~oda~~ ilerle yeni odaya

lamba açılma sırası

1→2→3 olur

②



returning phase

1- odaya ilerle

2- lambayı aç

lamba açılma sırası

1→2→3 ?

3-2-1 olur

```
void fun(int n){
        if(n>0){
Ascending     1. calling;
              2. fun(n-1)*2;
Descending    3. Returning;
        }
}
```

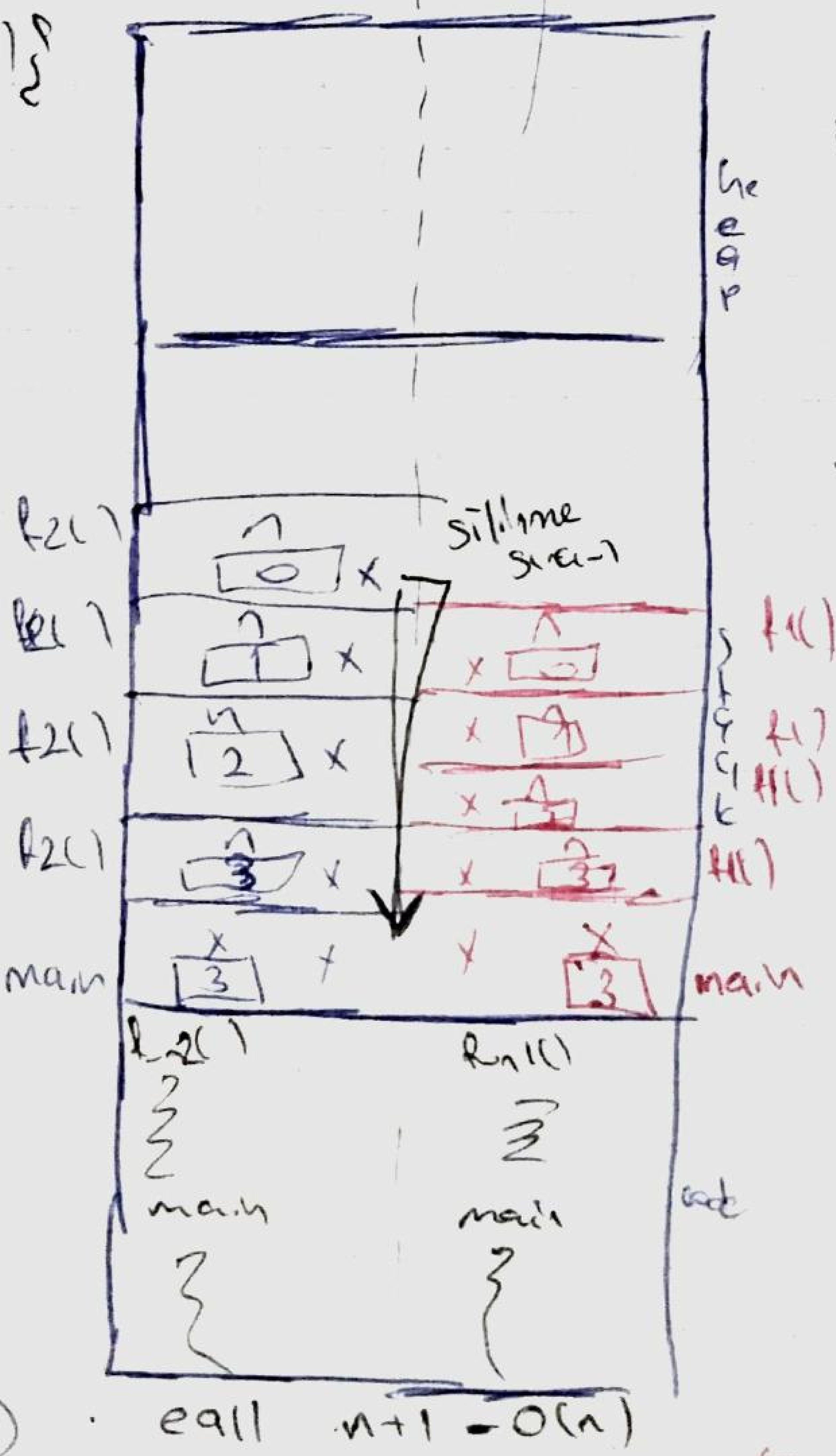
→ döngüler sadece ascending'e sahiptir

→ Recursiveler hem ascending'e hem de descending'e sahiptir

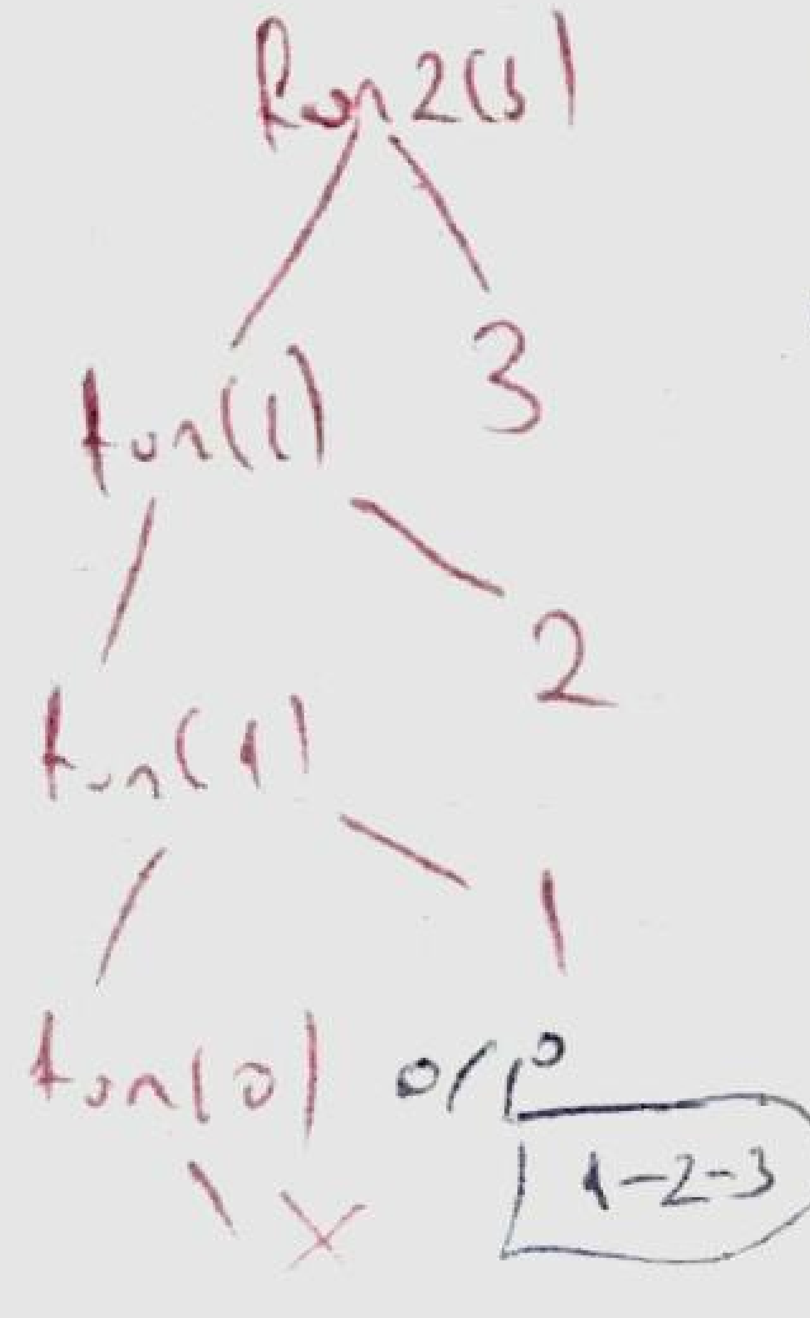
→ Zaten gördüğünde sondan alır.

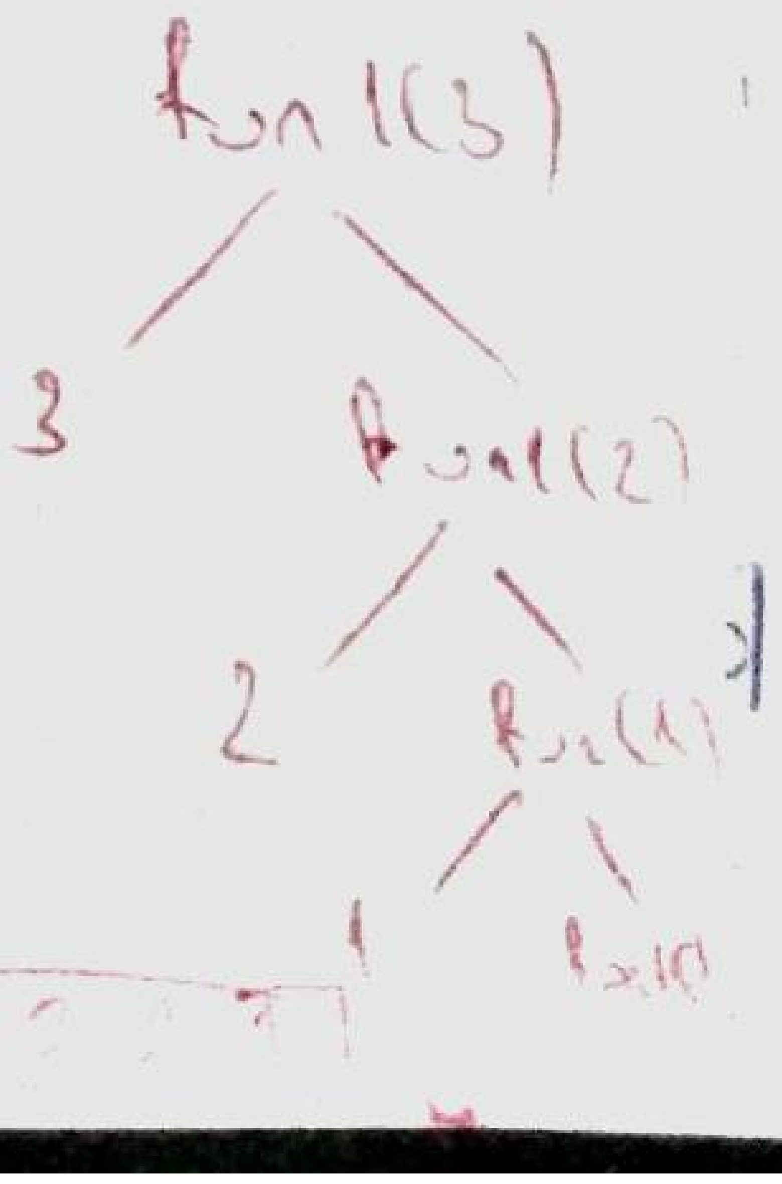
```
void fun2(int n){
  if(n>0){
    ~~cout<<n;~~
    fun2(n-1);
    cout<<n;
  }
}
void main(){
  int x=3
  fun2(x)
}
```

```
void fun1(int n){
if(n>0){
  cout<<n<<endl;
  fun1(n-1);
}
}
void main()
{
  int x=3;
  fun1(x);
}
```



call n+1 - O(n)

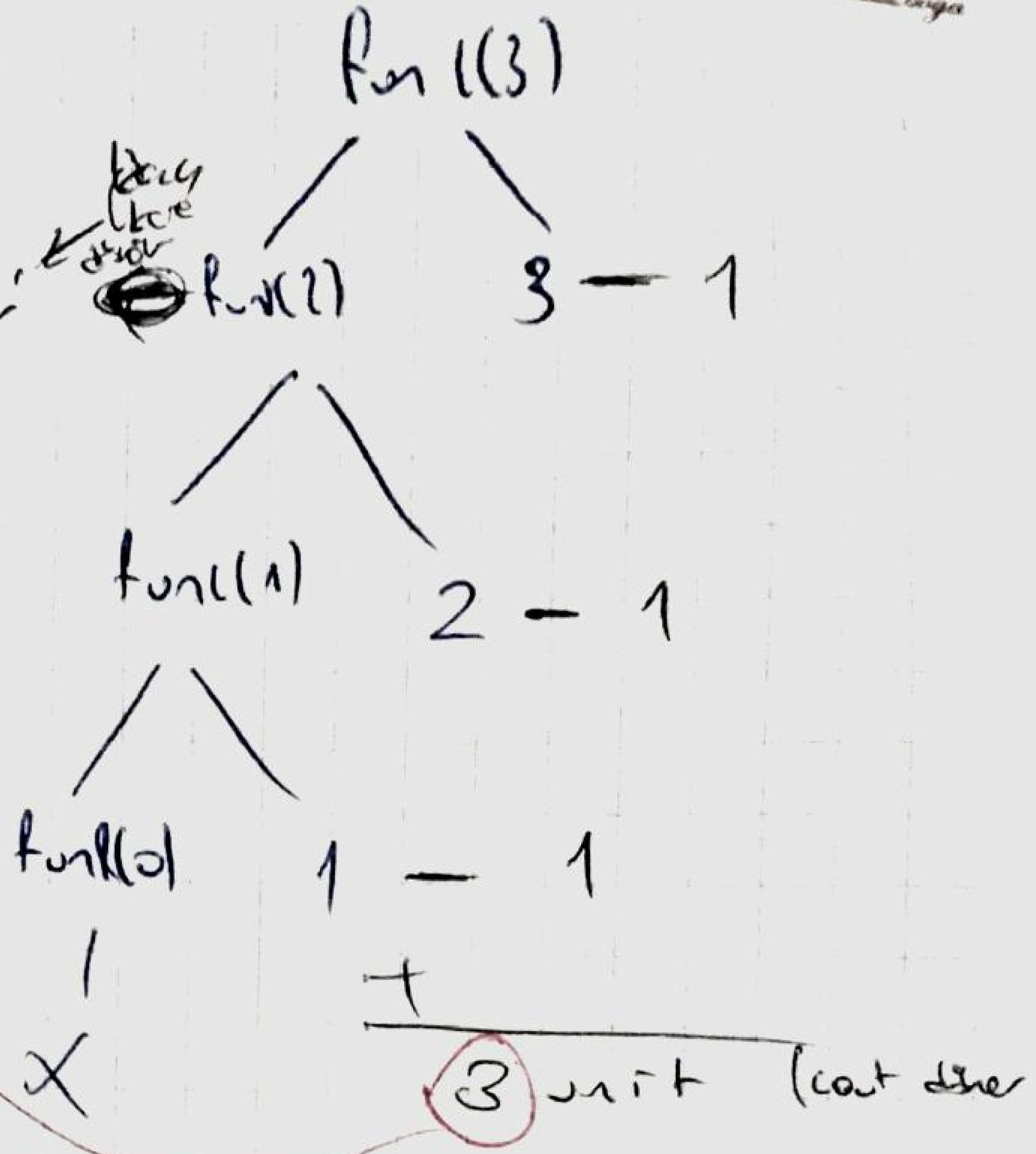




Recurrence Relation - Time Complexity of Recursive

```
void fun1(int n){
    if(n>0){
        cout<<n<<endl;
        fun1(n-1);
    }
}
```

```
int main(){
    int x=3
    fun(x);
}
```



n units

O(n)

```
void fun1(int n){
    if(n>0){
        cout<<n;
        fun1(n-1);
    }
}
```

kaç kere döner

→ T(n) olsun

→ 1

→ 1

→ T(n-1)

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

$$T(n)=\begin{cases}1 & n=0\\ T(n-1)+2 & n>0\end{cases}$$

1 yaptı

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

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

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

→ böyle devam eder

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

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

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

Assum $n-k=0$

$\therefore n=k$

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

$T(n)=T(0)+n$

$T(n)=1+n$

$-O(n)$