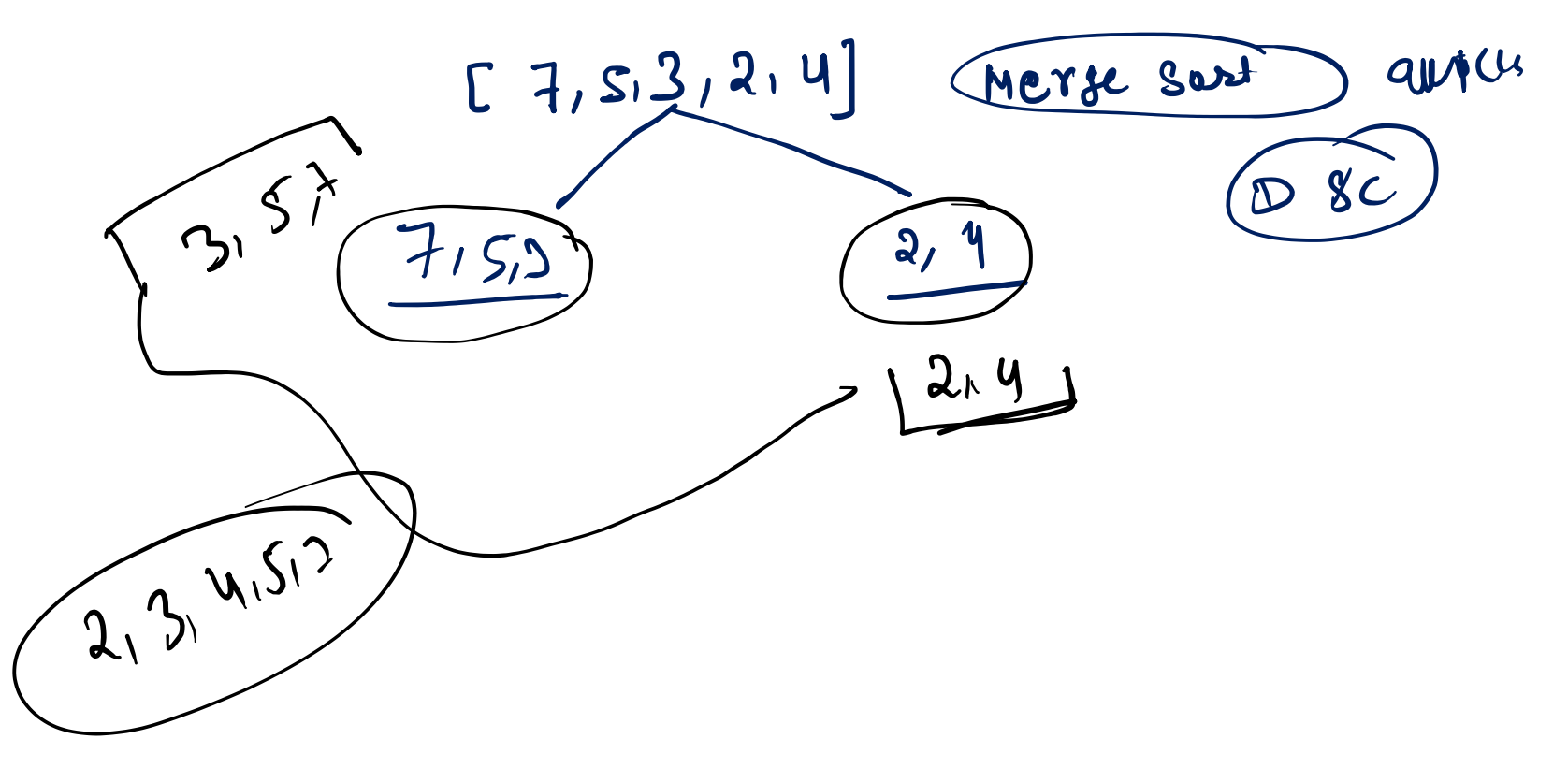
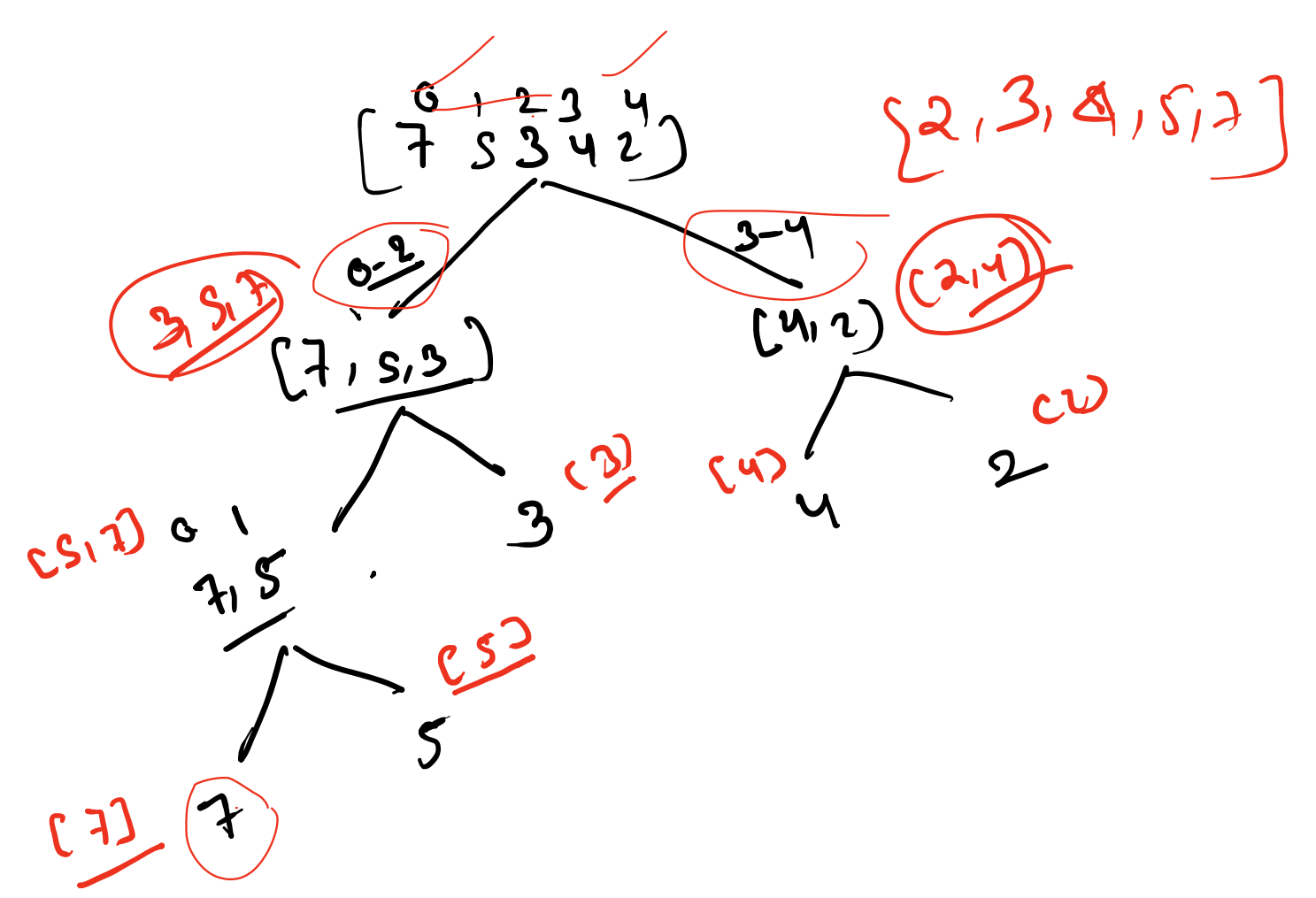
Lec-23

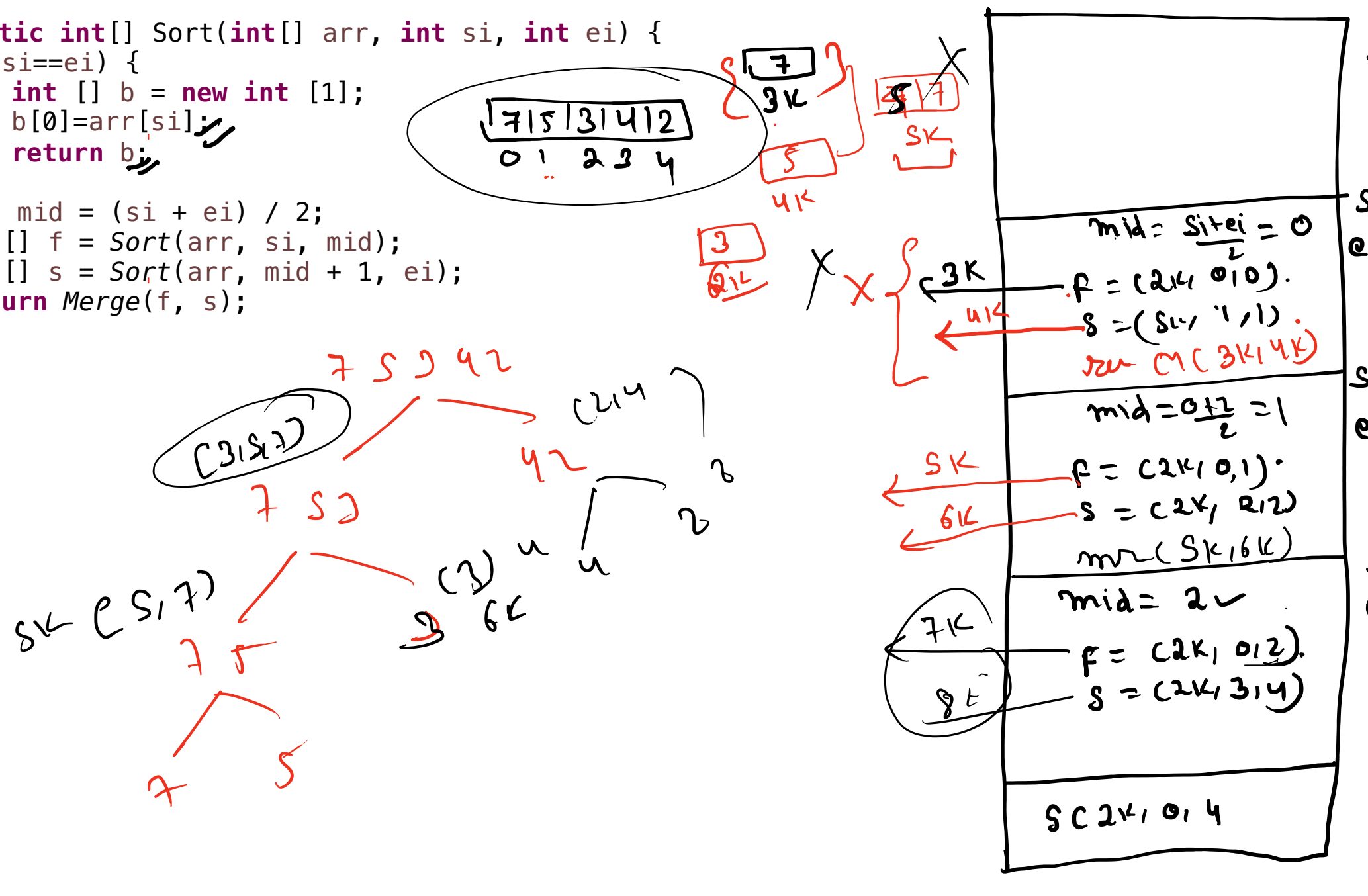
01/03/25 4:42 PM

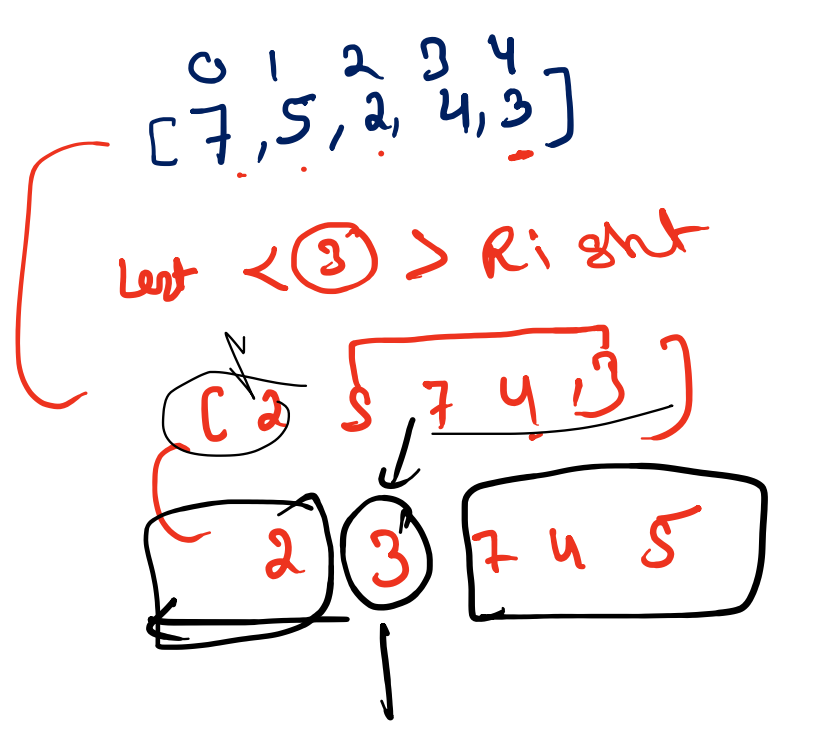




```
public static int[] Sort(int[] arr, int si, int ei) {
        if(si==ei) {
            int [] b = new int [1];
            b[0]=arr[si];
            return b;
        }
        int mid = (si + ei) / 2;
        int[] f = Sort(arr, si, mid);
        int[] s = Sort(arr, mid + 1, ei);
        return Merge(f, s);

    }
```

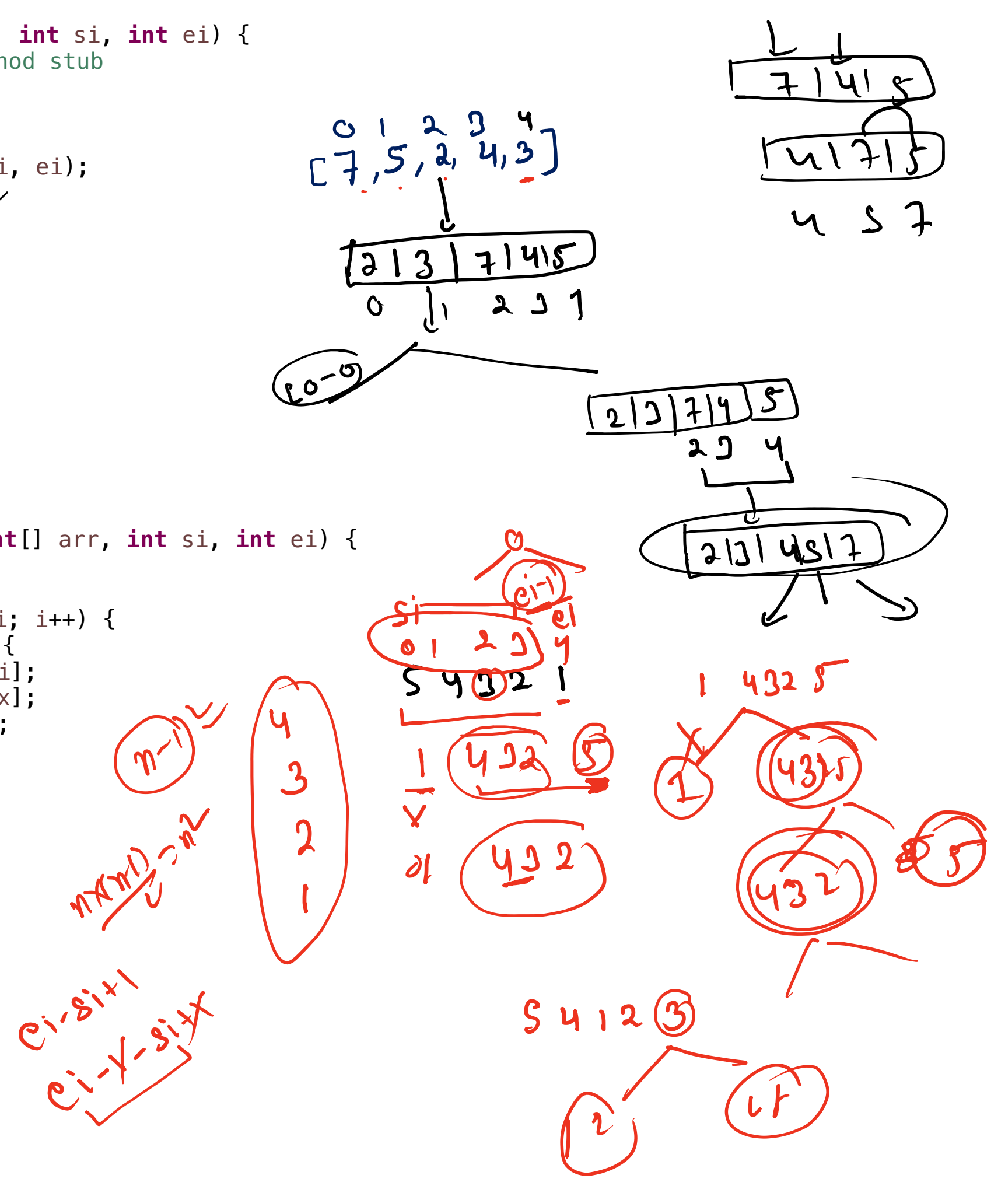




```
public static void sort(int[] arr, int si, int ei) {
        // TODO Auto-generated method stub
        if (si >= ei) {
            return;
        }
        int idx = Partition(arr, si, ei);
        sort(arr, si, idx - 1);
        sort(arr, idx + 1, ei);

    }
```

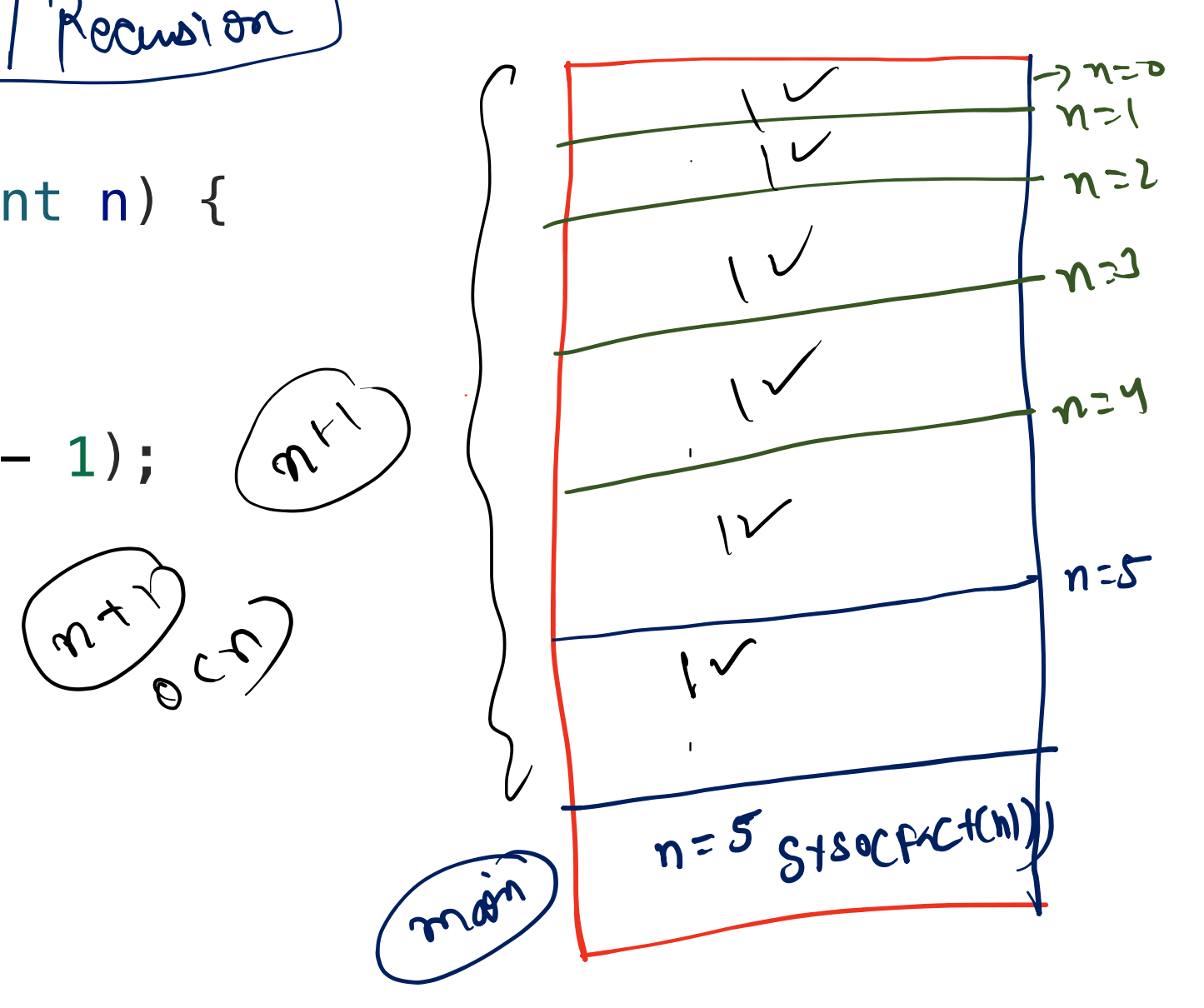
```
public static int Partition(int[] arr, int si, int ei) {
        int item = arr[ei];
        int idx = si;
        for (int i = si; i < ei; i++) {
            if (arr[i] < item) {
                int temp = arr[i];
                arr[i] = arr[idx];
                arr[idx] = temp;
                idx++;
            }
        }
        int temp = arr[ei];
        arr[ei] = arr[idx];
        arr[idx] = temp;
        return idx;
    }
```



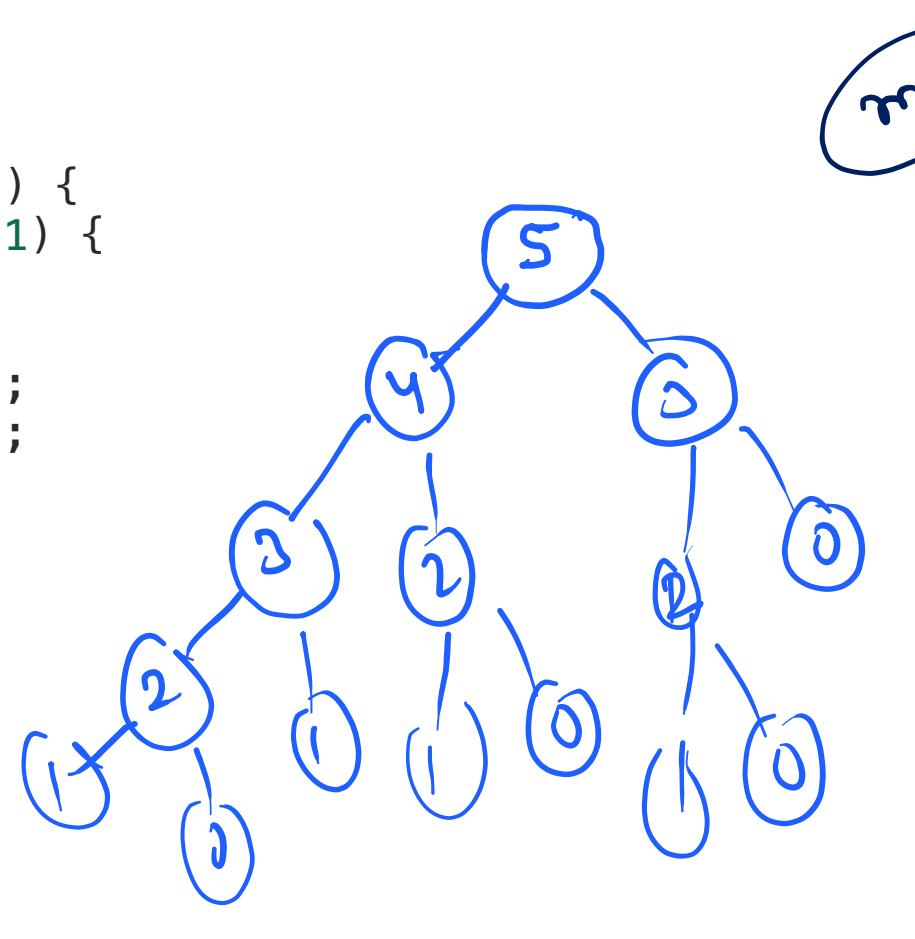
9:30 7:30

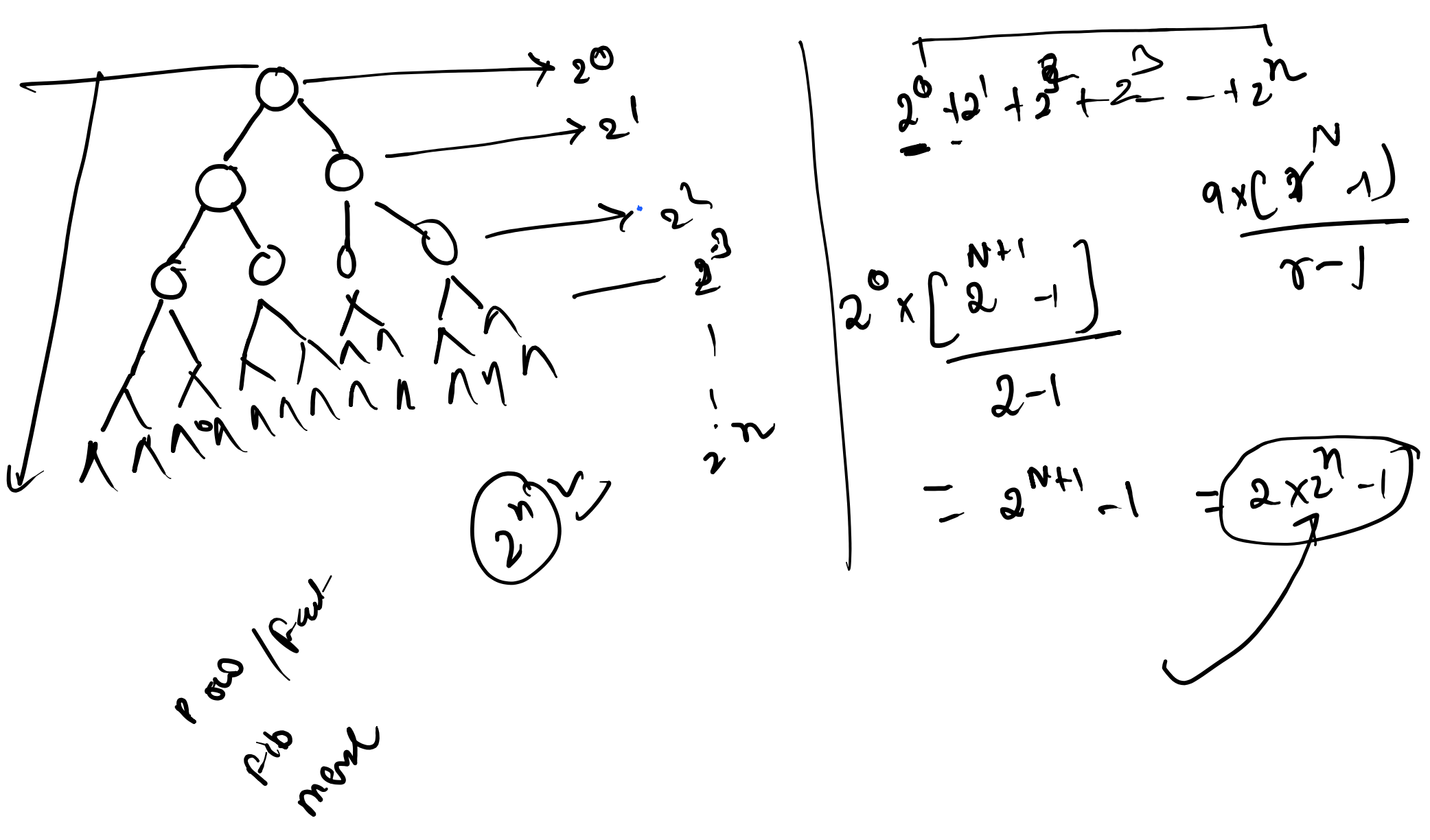
Recursion

```
public static int fac(int n) {
    if (n == 0) {
        return 1;
    }
    int fn = fac(n - 1);
    return fn * n;
}
```



```
public static int fib(int n) {
        if (n == 0 || n == 1) {
            return n;
        }
        int f1 = fib(n - 1);
        int f2 = fib(n - 2);
        return f1 + f2;
    }
```





```
public static int fac(int n) {
    if (n == 0) {
        return 1;
    }
    int fn = fac(n - 1);
    return fn * n;
}
```

$O(N)$

Time Solve

$T(N) = T(N-1)+1$
$T(N-1) = T(N-2)+1$
$T(N-2) = T(N-3)+1$
$T(N-3) = T(N-4)+1$
$T(N-4) = T(N-5)+1$
$T(N-5) = T(N-6)+1$

$T(0) = 1$

$T(n) = n$

$T(N) = T(N/2)+1$
$T(N/2) = T(N/2^2)+1$
$T(N/2^2) = T(N/2^3)+1$
$T(N/2^3) = T(N/2^4)+1$

$T(1) = 1$

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

$2^K = N$

$K = \log_2 n$

$T(n) = K$

$\log_2 n$

```
public static int fib(int n) {
        if (n == 0 || n == 1) {
            return n;
        }
        int f1 = fib(n - 1);
        int f2 = fib(n - 2);
        return f1 + f2;
    }
```

~~$T(N) = T(N-1)+T(N-2)+1$~~
~~$T(N-1) = T(N-2)+T(N-3)+1$~~
~~$T(N-2) = T(N-3)+T(N-4)+1$~~
~~$T(N-3) = T(N-4)+T(N-5)+1$~~
~~$T(N-4) = T(N-5)+T(N-6)+1$~~
~~$T(N-5) = T(N-6)+T(N-7)+1$~~
$T(0)$

$f(N) <= c\cdot g(N)$

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

$f(n)$ $c\,g(n)$

$T(n) = T(n-1)+T(n-1)+1$
$T(n) = 2T(n-1)+1$
$2T(n-1) = 2^2T(n-2)+1\times 2$
$2^2T(n-2) = 2^3T(n-3)+1\times 2^2$
$2^3T(n-3) = 2^4T(n-4)+1\times 2^3$
$2^4T(n-4) = 2^5T(n-5)+1\times 2^4$
$2^{n-1}T(1) = 1\times 2^{n-1}$

$T(n) = 2^0+2+2^2+\dots+2^{n-1}$
$= 2^n$

$2^0\times\frac{(2^n-1)}{2-1}$
$= 2^n-1$

$T(n) = T(n/2)+T(n/2)+N$
$T(N) = 2T(n/2)+N$
$2T(N/2) = 2^2T(n/2^2)+\frac{N}{2}\times 2$
$2^2T(n/2^2) = 2^3T(n/2^3)+\frac{N}{2^2}\times 2^2$
$2^3T(n/2^3) = 2^4T(n/2^4)+\frac{N}{2^3}\times 2^3$
$2^4T(n/2^4) = 2^5T(N/2^5)+\frac{N}{2^4}\times 2^4$
$2^KT(N/2^K=1) = \frac{N}{2^K}2^K$

$T(n) = K\cdot N$
$2^K = n$ $n\log n$
$\frac{N}{2^K} = 1$ $K =$
$2^K = N$
$K = \log n$