

## Power-logarithmic (Day 23)

$$\text{Power} : O(n)$$

$$O(\log(n)) \quad \text{no sub terms}$$

$$2^5 = 2 \times 2 \times 2 \times 2 \times 2$$

$$\text{Pow}(2, 5) \rightarrow \text{Pow}(2, 4) \rightarrow \text{Pow}(2, 3) \rightarrow \text{Pow}(2, 2) \rightarrow \text{Pow}(2, 1) \\ \downarrow \\ \text{Pow}(2, 0)$$

$$2^5 = \overset{2}{\boxed{2}} \times \overset{4}{\boxed{2 \times 2}} \times \overset{4}{\boxed{2 \times 2}}$$

$$\Rightarrow 2 \times \boxed{4 \times 4}$$

$$2 \times 2^2 \times 2^2$$

$$2 \times 2^4$$

$$3^5 \rightarrow \boxed{3} \times \boxed{3 \times 3} \times \boxed{3 \times 3}$$

$$\Rightarrow 3 \times 9 \times 9$$

$$2^4 \rightarrow \begin{array}{c} \boxed{2 \times 2} \times \boxed{2 \times 2} \\ 4 \times 4 \end{array}$$

$$2 \times \overset{32}{\underset{11}{4 \times 4}}$$

$$\xrightarrow{4} 2 \times 2$$

$$\xrightarrow{2} 2 \times 1 \times 1$$

$$\xrightarrow{1}$$

$$\text{Pow}(2, 5) \longrightarrow \text{Pow}(2, 2) \longrightarrow \text{Pow}(2, 1) \longrightarrow \text{Pow}(2, 0)$$

$$2 \times 2 \times 2 \times 2 \times 2$$

$$2 \times 2$$

$$\log(n)$$

$$2^5 \rightarrow 2 \times \overset{\times}{\underset{=}{\boxed{2 \times 2}}} \times \overset{\downarrow}{\boxed{2 \times 2}}$$

$$2^4 \rightarrow \boxed{2 \times 2} \times \boxed{2 \times 2}$$



2<sup>5</sup>

X

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

↑

$\text{pow}(2, 1) \rightarrow 1, 2$

↑

$\text{pow}(2, 3) \rightarrow 1, 2$

↑

$\text{pow}(2, 5) \rightarrow 1, 2$

$2 \times 1 \times 1$

②

$2 \times 2 \times 2 = 8$

```
public static int pow(int x, int y) {
    1 if (y == 0) {
        return 1;
    }

    2 int recAns = pow(x, y/2);

    3 if (y % 2 == 0) {
        3.1 return recAns * recAns;
    } else {
        4 return x * recAns * recAns;
    }
}
```

$$x^7 \rightarrow (x^{7/2}) * (x^{7/2})$$

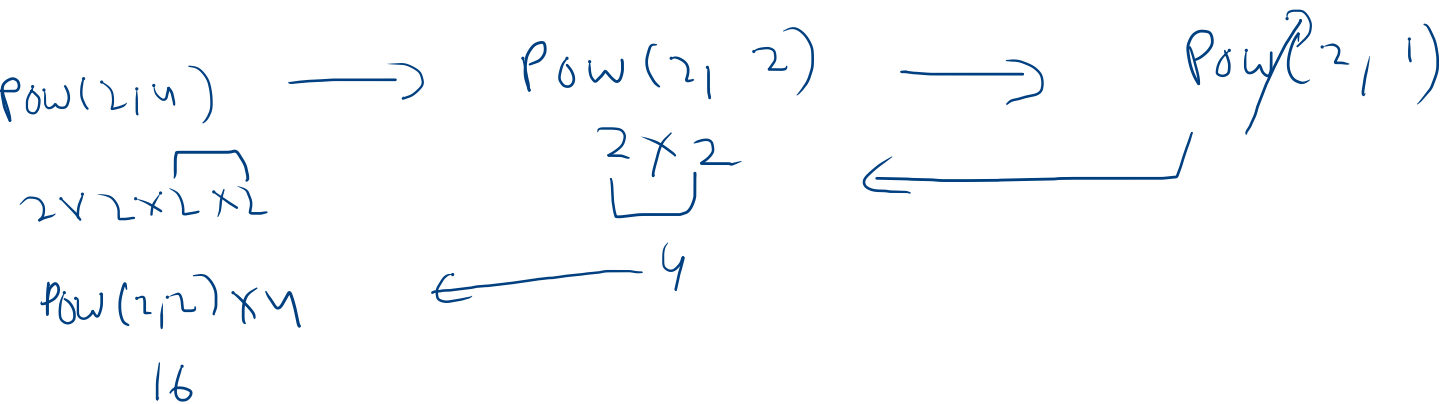
$$\downarrow \qquad \qquad \downarrow$$

$$x^{7/4} * x^{7/4} \qquad x^{7/4} * x^{7/4}$$

$$is(7000) \{$$

$$x * x * x * x\}$$

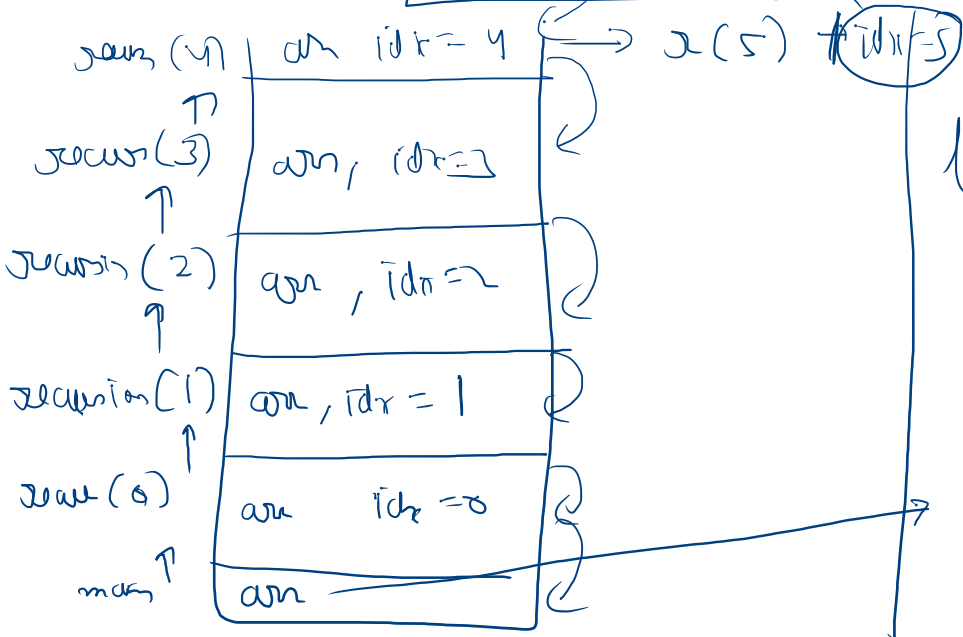
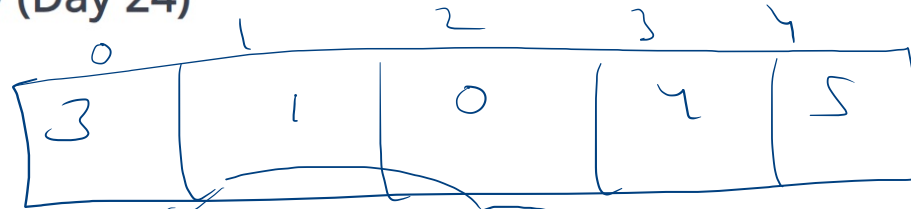
$$2^4 \rightarrow \sqrt{2 \times 2} \times \sqrt{2 \times 2}$$



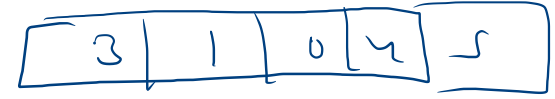
# Display Array (Day 24)

3  
1  
0  
4  
5

arr =

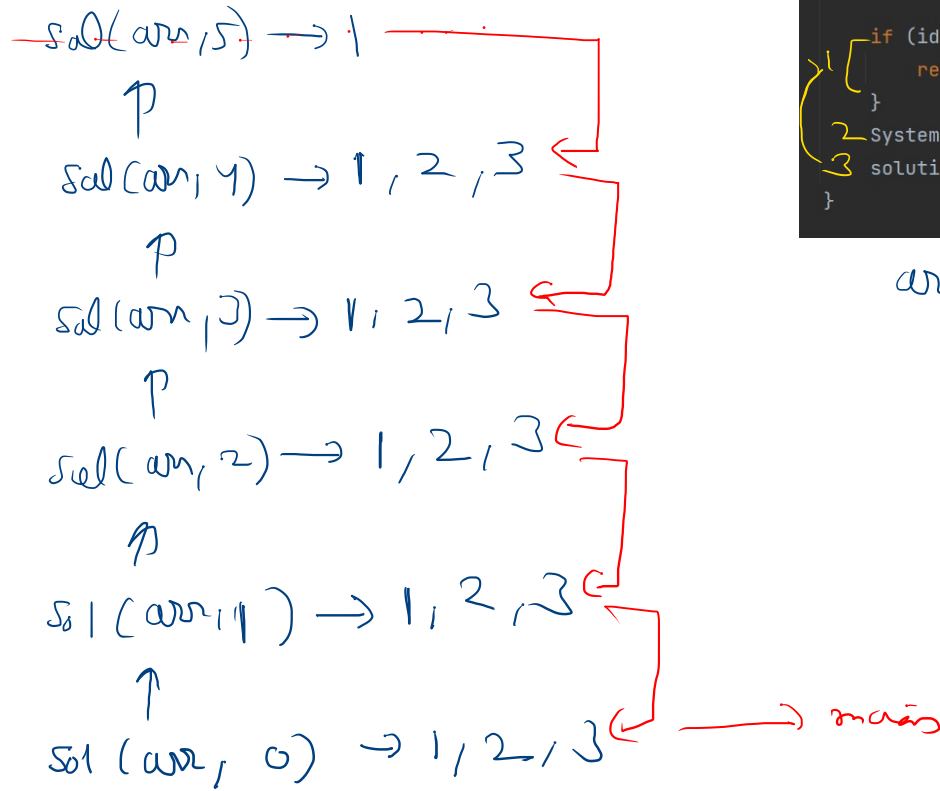


for (i = 0; i < arr.length; i++)





Recursive tree



```

2 usages
public static void solution(int[] arr, int idx) {
    1 if (idx == arr.length) {
        return;
    }
    2 System.out.println(arr[idx]);
    3 solution(arr, idx + 1);
}
  
```

arr → 1, 2, 3, 4, 5

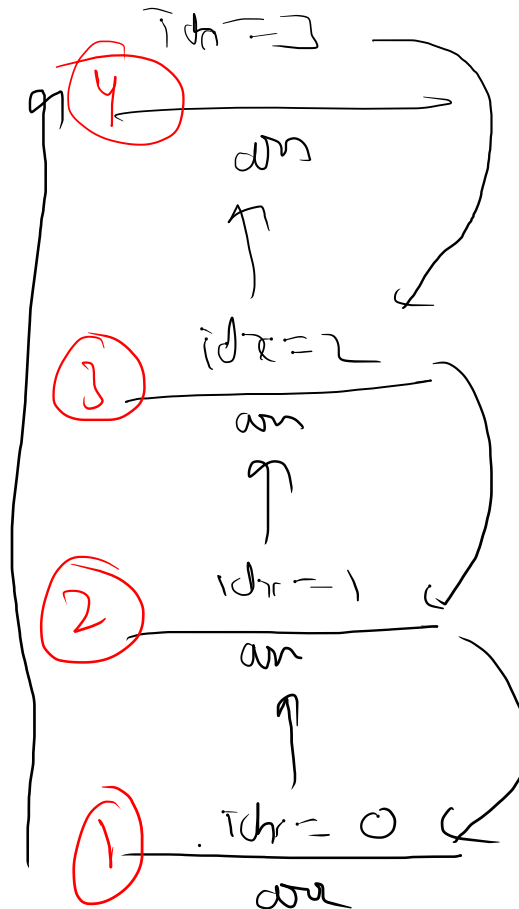
len = 5

1	5
2	4
3	3
4	2
5	1



# Recursive Tree

Recursive  
Stack  
fills



Clear  
your stack

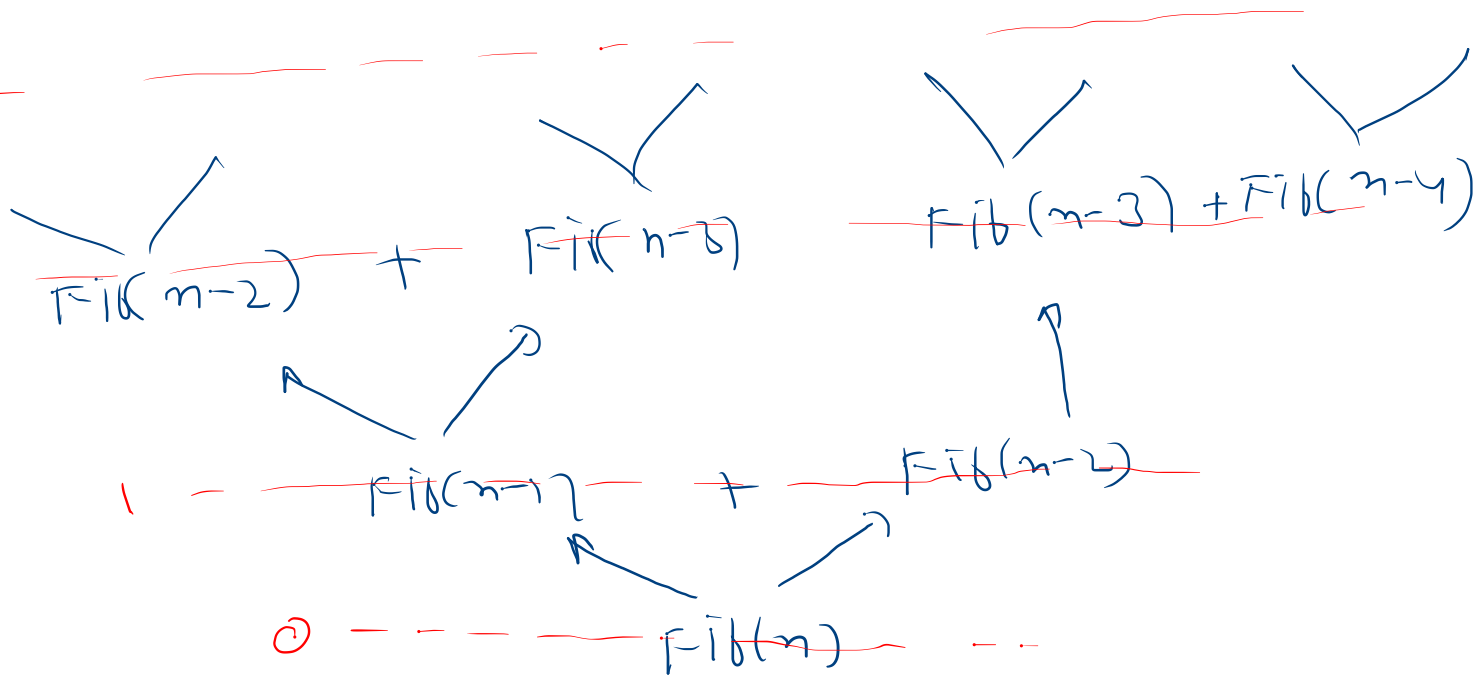
$\text{Fib}(n)$       1      1      2      3      5      8      13      21  
 $n =$  1      2      3      4      5

$$\text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2)$$

$$\text{Fib}(5) = \text{Fib}(4) + \text{Fib}(3)$$

3      + 2      = 5

Recursive  
tree



## First Index (Day 24)

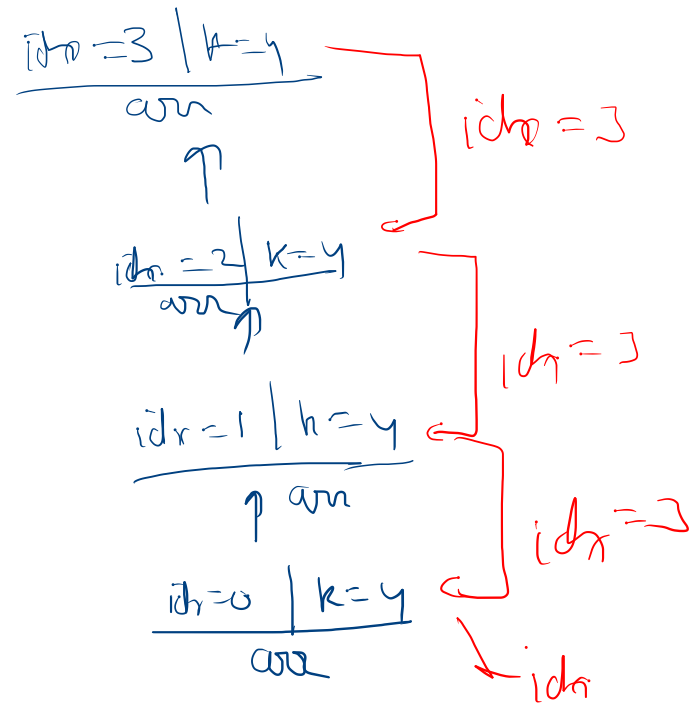
	0	1	2	3	4	5
arr =	15	11	40	4	4	9

key = (4) ↗

o/p = 3

key = (10) o/p = -1

	0	1	2	<u>3</u>	4	5
arr =	15	11	40	4	4	9



	0	1	2	3	4	5
arr-	15	11	40	4	4	9

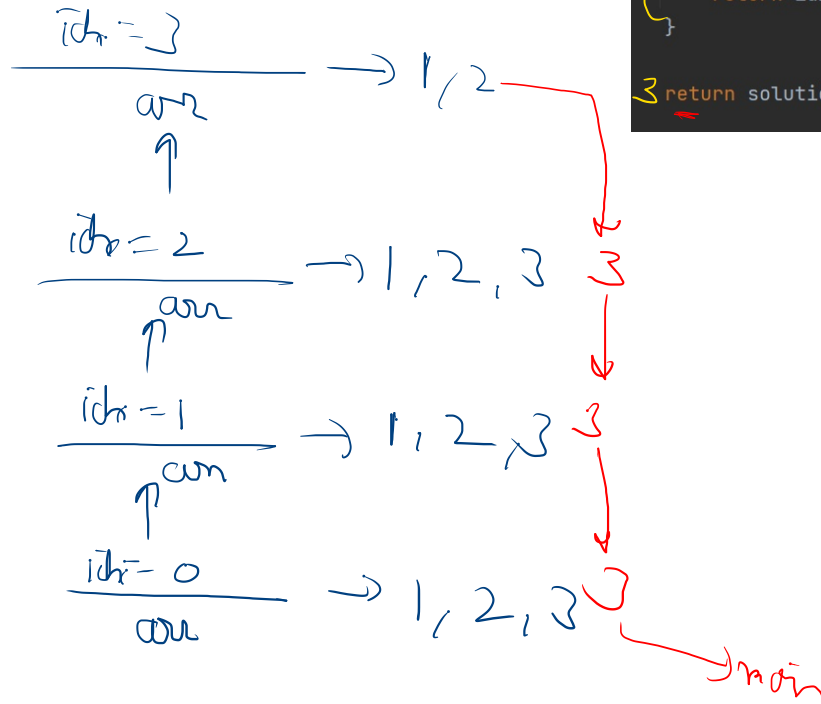
$k=10$

$$\begin{array}{r} 6 \\ \hline arr \end{array}$$

$$\begin{array}{r} 5 \\ \hline arr \\ \uparrow p \\ 4 \\ \hline arr \\ \uparrow p \\ 3 \\ \hline arr \\ \uparrow p \\ 2 \\ \hline arr \\ \uparrow p \\ 1 \\ \hline arr \\ \uparrow p \\ r_d = 0 \\ \hline arr \end{array}$$

	0	1	2	3	4	5
arr =	15	11	40	4	4	9

key = 4



```

1 if (idx == arr.length) {
2   return -1;
3 }
4
5 if (arr[idx] == key) {
6   return idx;
7 }
8
9 return solution(arr, idx + 1, key);

```



## All Indices Of Array (Day 24)

	0	1	2	3	4	5
arr =	15	11	40	4	4	9

$$\underline{K = 4}$$

$$\text{O/P} = [3, 4]$$

journaal  $\rightarrow$  AL  $<$



