

Backtracking \rightarrow Find solution by exploring all possibilities/combinations

```
fun(N) {  
  if (N == 0)  
    ret 0  
  else  
    ret fun(N/2) * 10 + N % 2  
}  
fun(7)
```

$$f(0) = 0$$

$$f(3)$$

$$f(1) \times 10 + 3 \% 2 \\ 1 \times 10 + 1 \\ = 11$$

$$f(3) \times 10 + 7 \% 2 \\ 11 \times 10 + 1 \\ = 111$$

$$f(7)$$

N
 $N/2$
 $N/4$
 $N/8$
 \vdots
 1
 0

$$\log(n)$$

Quiz TC

$$\log(n)$$

Quiz 2

```
void fun (char[] s, int x) {
```

```
    print(s)
```

```
    char temp
```

```
    if (x < s.len() / 2) {
```

```
        temp = s[x]
```

```
        s[x] = s[s.len() - x - 1]
```

```
        s[s.len() - x - 1] = temp
```

```
        fun(s, x + 1)
```

```
    }
```

```
}
```

LLORCS ✓
fun ("SCROLL", 0)
temp = s

SCROLL
LCROLS
LLROLS
LLORCS

take some elem / leave other elem
↑

Q1 Generate all subsets of given array

eg1 $\{10, 20, 30\} \rightarrow \{\}$

All subarrays are subset

$\{10\}, \{20\}, \{30\}$

All subset NOT subarray

$\{10, 20\}, \{10, 30\}, \{20, 30\}$

$\{10, 20, 30\}$

All possible subsets using recursion

$1 \rightarrow 2$

$2 \rightarrow 4$

$3 \rightarrow 8$

Params \rightarrow

1) $i \rightarrow$ current idx

2) $\text{list} \langle \text{int} \rangle \text{ ans} \rightarrow$ subset

$\{10, 20, 30\}$

$0, \{\}$

take \swarrow leave \searrow

$1, \{10\}$

$1, \{\}$

$\swarrow \searrow$
 2

$\swarrow \searrow$
 $\{\}$

$2, \{10, 20\}$

$2, \{10\}$

$2, \{20\}$

$2, \{\}$

$\swarrow \searrow$
 $3, \{10, 20, 30\}$

$\swarrow \searrow$
 $3, \{10, 20\}$

$\swarrow \searrow$
 $\{\}$

$\swarrow \searrow$
 $3, \{10, 30\}$

$\swarrow \searrow$
 $\{20\}$

$\swarrow \searrow$
 $3, \{20, 30\}$

$\swarrow \searrow$
 $\{10\}$

$\{10, 30\}$

```

void subset (int[] arr, int idx, list<int> ans) {
    if (idx == n) {
        print (ans)
        return;
    }
}

```

{10}

// include

```

ans.push (arr[idx])
subset (arr, idx+1, ans)
ans.remove()

```

{10, 20}

// exclude

```

subset (arr, idx+1, ans)

```

}

TC: 2^n subsets

max size of a subset = n

TC: $O(N 2^n)$

⁰10 ¹20 ²30

0, (2) → 2 3
↙ ↘

1, {10} → 10

1, {3}

↙ ↘

↙

2, {10, 20}

2, {10}

2, {20}

↙ ↘

↙

3, {10, 10, 10} 3, {10, 20}

3, {20, 10}

arrangement
↑

Q2 Generate all permutation of array with distinct elements.

Eg- 4 7 9

1 → 1

2 → 2

3 → 6

4 → 24

5 → 120

⋮

4 7 9

4 9 7

7 4 9

7 9 4

9 4 7

9 7 4

4! = 24

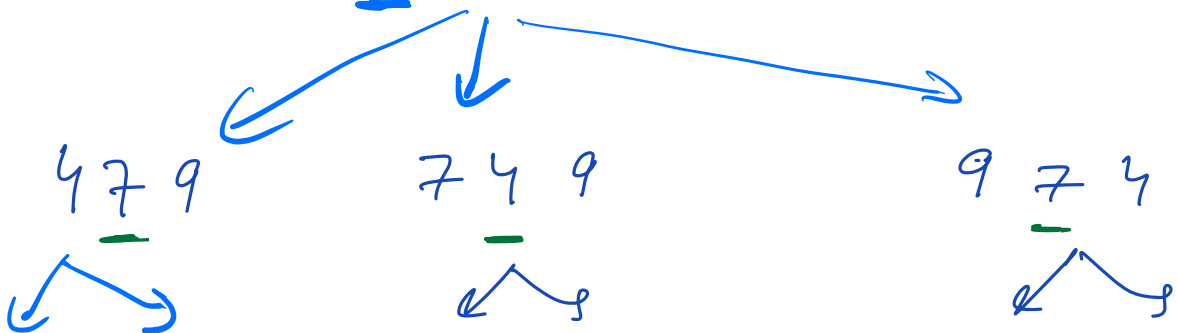
5! = 120

n → factorial (n)

n!

i = 0

4 7 9



4 7 9

4 9 7

7 4 9

7 9 4

9 7 4

9 4 7

parameters \rightarrow

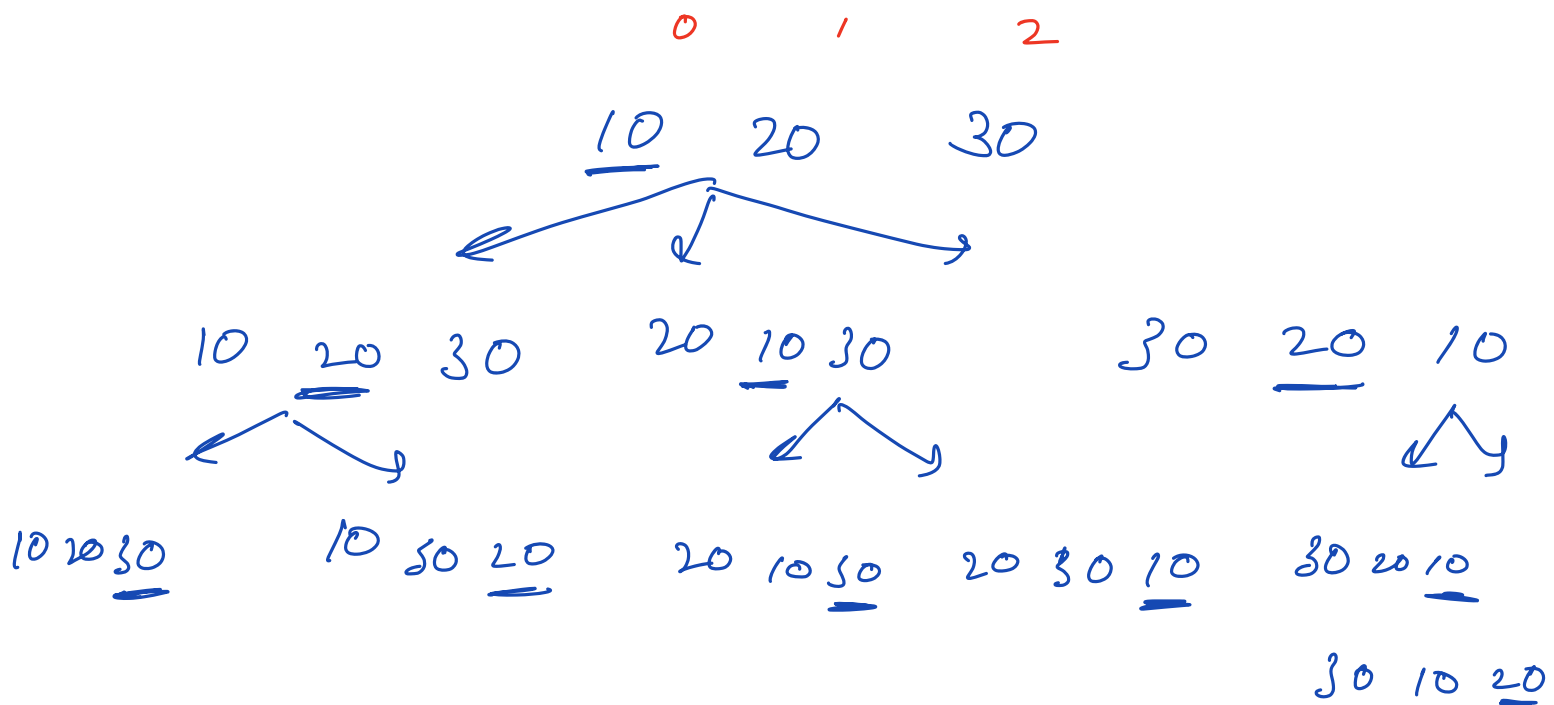
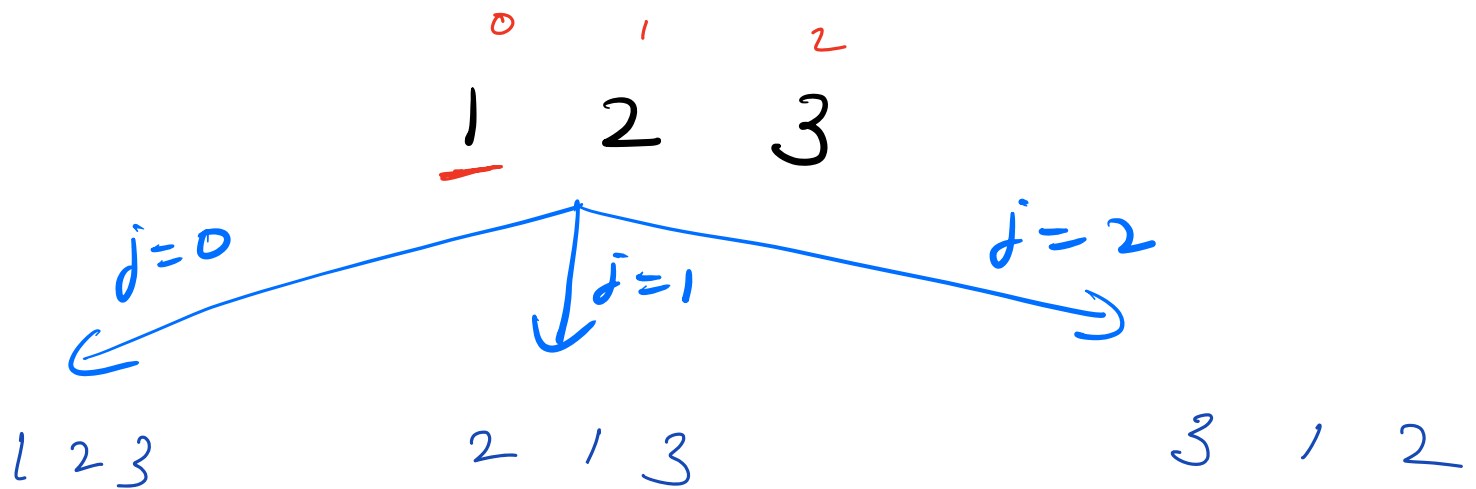
1) $i \rightarrow$ index

```
void perm (int[] arr, int i) {  
    if (i == n-1) {  
        print(arr)  
        return;  
    }  
    for (j=i; j < n; j++) {  
        swap(a[i], a[j])  
        perm(i+1)  
        swap(a[i], a[j])  
    }  
}
```

TC:

SC:

If we don't swap back



of permutations = $n!$

Size of any perm = n

TC: $O(n * n!)$


```

void perm (int[] arr, int i) {
    if (i == n-1) {
        print(arr);
        return;
    }
    for (j=i; j<n; j++) {
        swap(arr[i], arr[j]);
        perm(i+1);
        swap(arr[i], arr[j]);
    }
}

```

