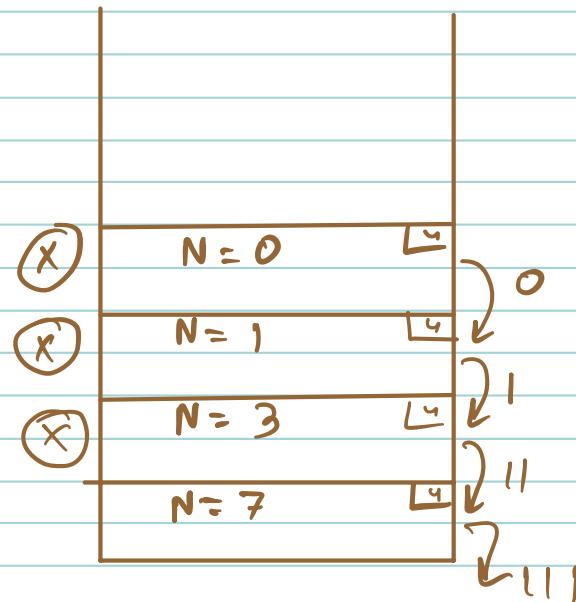


Backtracking

Ques! what is the output of below code for $N=7$?

```
int magicfun (int N) {  
    1     if (N==0) {  
    2         return 0;  
    3     } else {  
    4         return magicfun (N/2)*10 + (N%2);  
    5     }  
}
```



Ans = 111

$T.C \rightarrow$ No of function calls * Time taken by one fun call
 $O(C \log N)$

$S.C \rightarrow O(C \log N)$

Quiz 2 what will be the output of below code?

$s = ['s', 'c', 'r', 'o', 'l', 'l']$

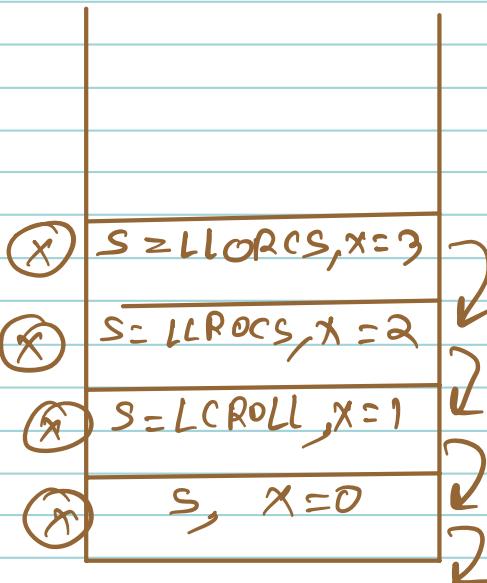
$\Rightarrow \boxed{\text{fun}(s, 0);}$; calling Function.

Code:-

```
void fun( char s[], int x) {  
    1     Print(s)  
    2     char temp;  
    3     if (x < s.length / 2) {  
        4         temp = s[x]  
        5         s[x] = s[s.length - x - 1];  
        6         s[s.length - x - 1] = temp;  
        7         fun(s, x + 1);  
    }  
}
```

Output

- $\rightarrow \text{SCROLL}$
- $\rightarrow \text{LCROLS}$
- $\rightarrow \text{LLR OCS}$
- $\rightarrow \text{LL ORCS}$



SCROLL
LCROLS
LLR OCS

LL ORCS

$T.C \rightarrow O(N/2) \approx O(N) \mid SC \rightarrow O(N)$

question :- Print all the Subsets

→ Need not to be Continuous

Ex:- arr[] = { 1 2 3 4 }

↳ All Subsets :-

{ }
{ 1 }
{ 2 }
{ 3 }
{ 4 }
{ 1 2 }
{ 1 3 }
{ 1 4 }
{ 2 3 }
{ 2 4 }
{ 3 4 }
{ 1 2 3 }
{ 1 2 4 }
{ 1 3 4 }
{ 2 3 4 }
{ 1 2 3 4 }

Ex2:- arr[] = { 1, 2, 3, 4 }

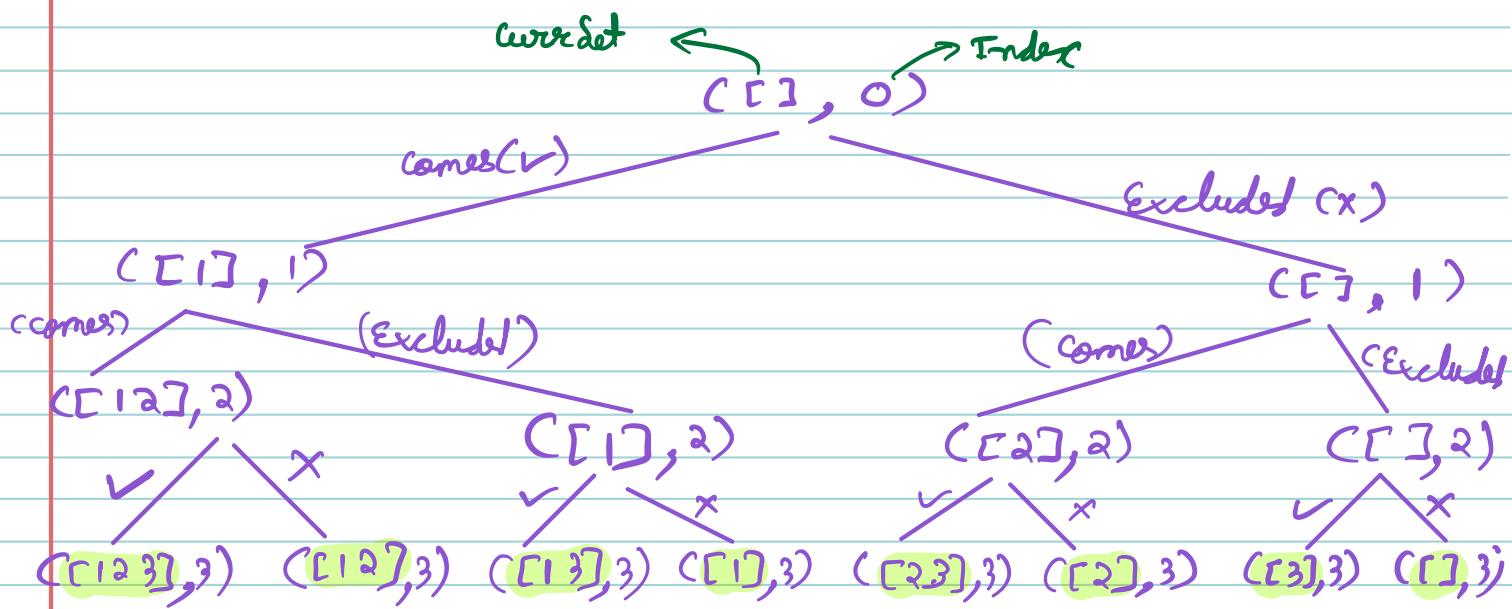
↳ All Subsets :-

{ }
{ 1 }
{ 2 }
{ 3 }
{ 4 }
{ 1 2 }
{ 1 3 }
{ 1 4 }
{ 2 3 }
{ 2 4 }
{ 3 4 }
{ 1 2 3 }
{ 1 2 4 }
{ 1 3 4 }
{ 2 3 4 }
{ 1 2 3 4 }

⇒ $2^n \rightarrow$ Count of Total Subsets.

Observation / Solution

$\text{arr}[] = \{1, 2, 3\}$



PSEUDO CODE

```
Void subsets ( int[] arr, int idx, List<Int> curSet ) {
    if ( idx == arr.length ) {
        Print ( curList );
        return;
    }
    // Pick
    curSet.add ( arr[idx] );
    subsets ( arr, idx + 1, curSet );
    curSet.remove ( curSet.size() - 1 );
}
// Don't Pick
subsets ( arr, idx + 1, curSet );
```

** → \star

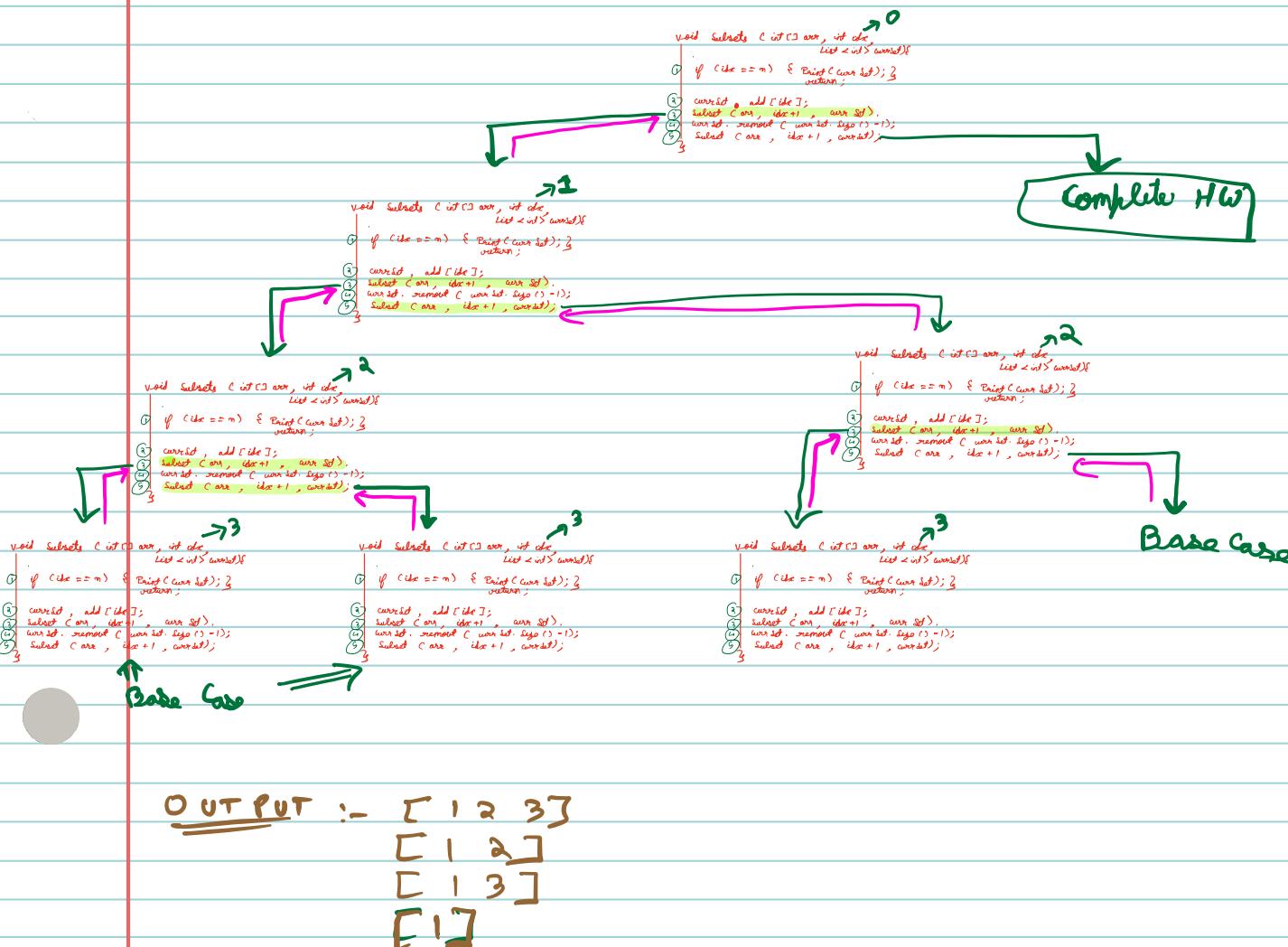
2

Print (curList);
return;
→ ans.add (curList);
ans.add (copy of
curSet);

look

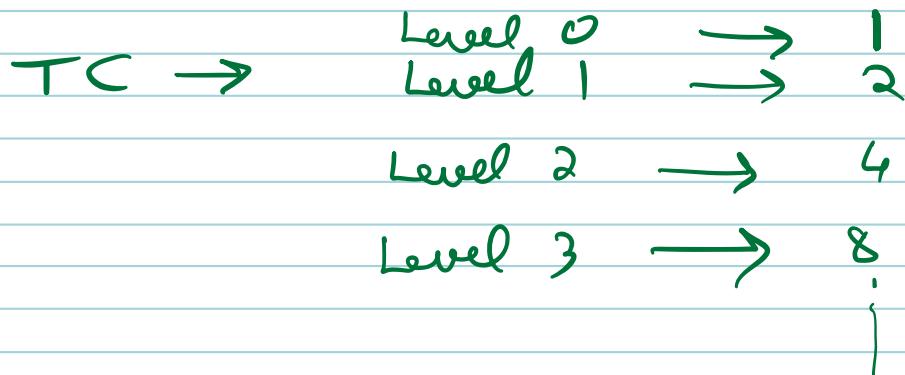
DRY RUN

$\text{arr} = [1 \ 2 \ 3]$
 $\text{curr set} = []$



OUTPUT :-

- [1 2 3]
- [1 2]
- [1 3]
- [2 3]
- [1]
- []



$$N = 3 , \quad 2^0 + 2^1 + 2^2 + 2^3$$

↓ Generic

$$N , 2^0 + 2^1 + 2^2 + 2^3 + \dots + 2^h$$

$T.C \rightarrow O(2^n) | SC \rightarrow O(N)$

$\text{Ans} = [\quad]$

$\text{Ans} = [\quad]$

$\boxed{\text{Break : - } 10:31 - 10:41}$

Question :- Given a string with Distinct characters, print all string permutations.

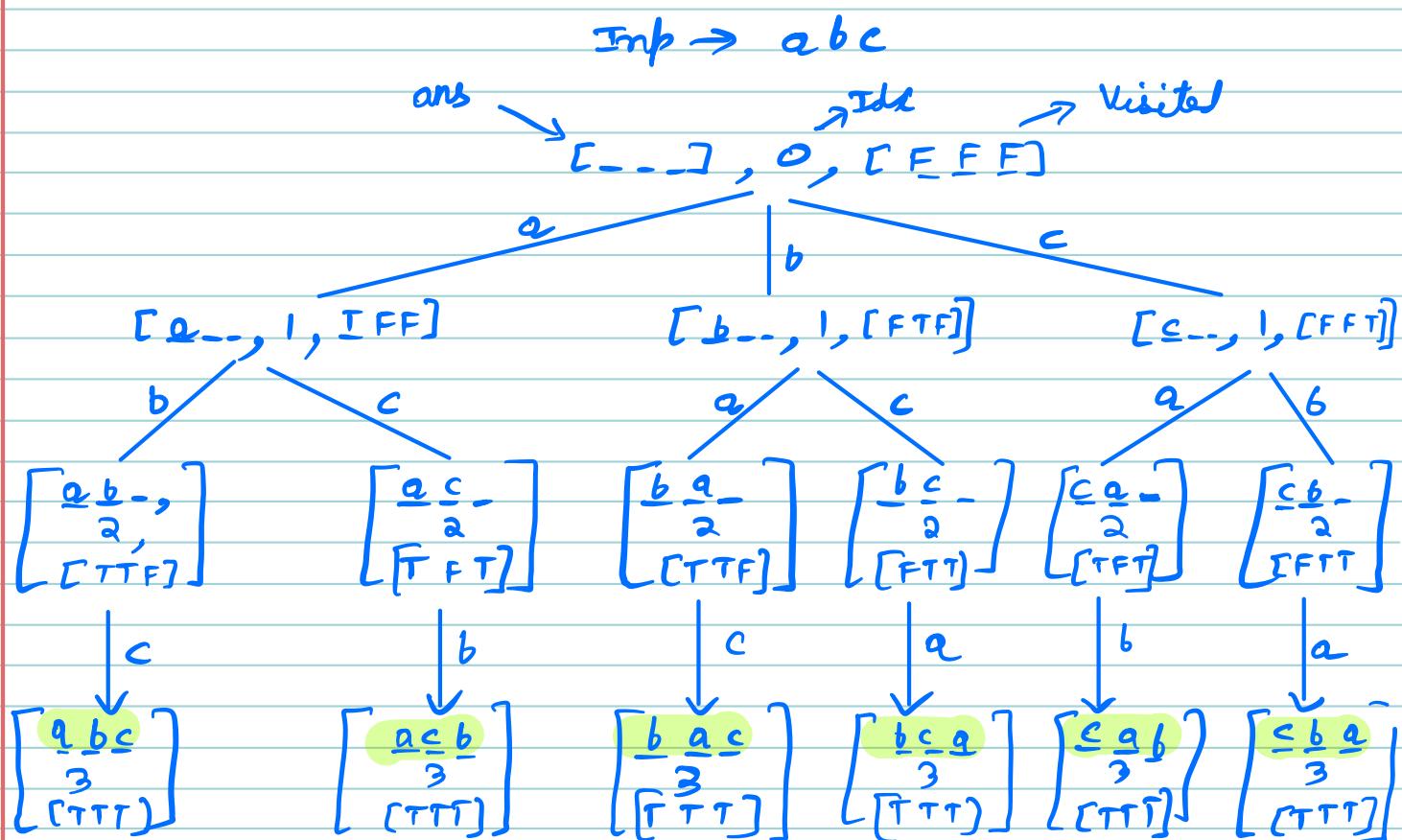
$\boxed{abc} \rightarrow$

$a b c$	$a c b$	$b c a$	$b a c$	$c a b$	$c b a$
---------	---------	---------	---------	---------	---------

[lowercase letter only]

if n , $n!$ permutations

Explanation



PSEUDO CODE

String str,
void permutations (char ans[], int idx, visited[]){
 if (idx == str.length()) {
 Print (ans); return;
 }
 for (int i = 0; i < n - 1;) {
 if (vis[i] == False) {
 ans[idx] = str[i];
 vis[i] = True;
 Permutation (ans, idx + 1, vis);
 vis[i] = False
 }
 }
}

Stupidity in above Code :- Revert the changes
made before going to next iteration.

$$T_C \rightarrow O(N * N!)$$

Work done to
Reach base
case

No. of base cases

$$S_C \rightarrow O(N)$$

⇒ DOUBT SESSION

Question :- K^{th} symbol Hard.

$0 \rightarrow 01$
$1 \rightarrow 10$

$n=1$

0

$n=2$

0 1

$n=3$

0 1 1 0

$n=4$

0 1 1 0 1 0 0 1

$n=5$

0 1 1 0 1 0 0 1 1 0 0 0 1 0 1 1 0

⇒ if ($n=5$, $K=3$) → [0 Ans]

⇒ if ($n=5$, $K=7$) → [1 Ans]

Easy Version :- you can generate the required level string & tell ans.

Hard Version :- easy approach gives TLE

↳ {Constraints}

Main challenge.

Observation

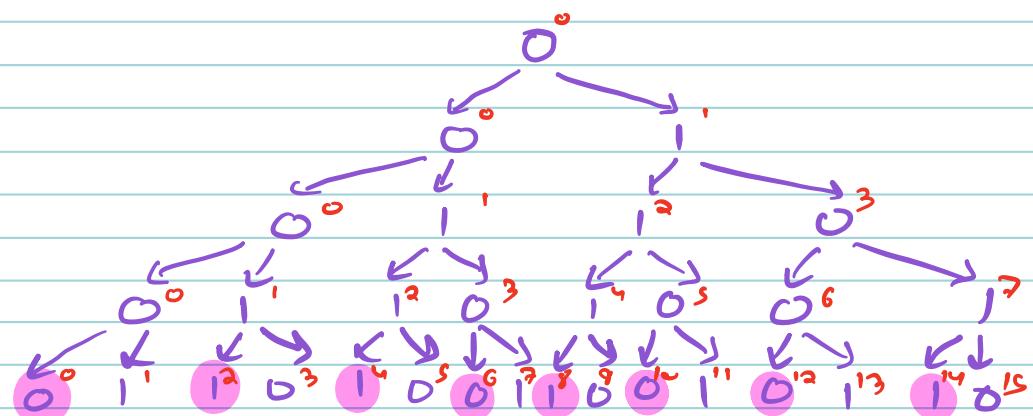
$n=1$

$n=2$

$n=3$

$n=4$

$n=5$



n, k

Observation

↳ Every even index element is same as its parent.

↳ Every odd index element is opposite from its parent.

↳ parent Index: $\frac{\text{My Index}}{2}$

↳ if $k=0$, Ans: 0 || if $n=0$, then Ans: 0

Base Case

Dry Run

Ex:- $(n=5, k=6)$

$0 \uparrow \downarrow$

$0 \underline{\text{Ans}}$

↳ Parent

$(n=4, k=3)$

$0 \uparrow \downarrow$

↳ Parent

! $(n=3, k=1)$

$1 \uparrow \downarrow$

↳ Parent

! $(n=2, k=0)$

0

Ex2 :-

$(n=5, k=7) \rightarrow 1 \underline{\text{Ans}}$

$1 \uparrow \downarrow$

! $(n=4, k=3)$

$0 \uparrow \downarrow$

! $(n=3, k=1)$

! $(n=3, k=0)$

$1 \uparrow \downarrow$

PSEUDO CODE

```
int  $k^{th}$  symbol (  $N, k$  ) {  
    Base Case →  
    if (  $N == 0 \text{ || } k == 0$  ) return 0;  
    if (  $k \% 2 == 0$  ) {  
        | return  $k^{th}$  symbol (  $N-1, k/2$  );  
    } else {  
        | return 1 -  $k^{th}$  symbol (  $N-1, k/2$  );  
    }  
}
```

3

$$TC \rightarrow O[\min(n, \log k)]$$

$$\downarrow 2^{n-1}$$

$$SC \rightarrow O(N)$$