

Ques 2)

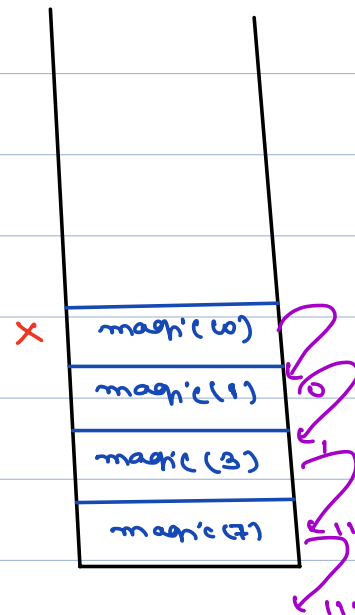
```
int magicfun( int N) {
    if ( N == 0)
        return 0;
    else
        return magicfun(N/2) * 10 + (N % 2);
}
```

$N = 7$

Soln :-

T.C  $\rightarrow O(\log n)$

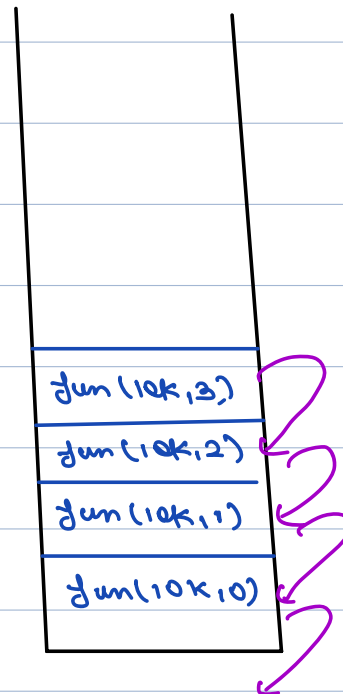
S.C  $\rightarrow O(\log n)$



Ques 2)

```
1. void fun(char s[], int x) {
2.     System.out.println(s);
3.     char temp;
4.     if(x < s.length/2) {
5.         temp=s[x];
6.         s[x] = s[s.length-x-1];
7.         s[s.length-x-1]=temp;
8.         fun(s, x+1);
9.     }
10. }
```

Run for fun("SCROLL", 0);

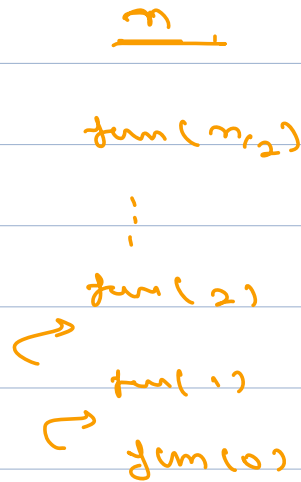


10k, ~~S~~ ~~C~~ ~~R~~ ~~O~~ ~~L~~ ~~L~~  
L O R C S

SCROLL  
LEAOLS  
LHROCS  
LHORCS

T.C  $\Rightarrow O(2^n)$

S.C  $\Rightarrow O(1)$



Ques)

arr[] = 2, 1, 2, 3, 3,

2, 1, 3, 3

2, 1, 2, 3, 3

Print all subsets.

2, 1, 2, 3, 4, 3 →

2, 3

2, 1, 3

2, 2, 3

2, 3, 3

2, 4, 3

2, 1, 2, 3

2, 1, 3, 3

2, 1, 4, 3

2, 2, 3, 3

2, 2, 4, 3

2, 3, 4, 3

2, 1, 2, 3, 3

2, 1, 3, 4, 3

2, 2, 3, 4, 3

2, 1, 2, 4, 3

2, 1, 2, 3, 4, 3

subsets.

$\{1, 2, 2'\}$   $\rightarrow$

$\downarrow$

$2^n$ .

(for count of subsets)

$\{1, 2, 2'\}$   
 $\{1, 2\}$

$\{2, 2'\}$

$\{2, 2'\}$

$\{1, 2\}$

$\{2, 2'\}$

$\{1, 2'\}$

$\{1, 2, 2'\}$

Ques)

arr  $[ ] = \{10, 20, 30\}$

Print all subsets.

8 subsets

$\{1\}$

$\{10\}$

$\{20\}$

$\{30\}$

$\{10, 20\}$

$\{10, 30\}$

$\{20, 30\}$

$\{10, 20, 30\}$

subset, idx

$([ ], 0)$

$\checkmark$

$\times$

$([10], 1)$

$([ ], 1)$

$\checkmark$

$\times$

$\checkmark$

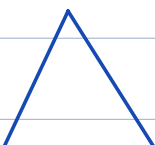
$\times$

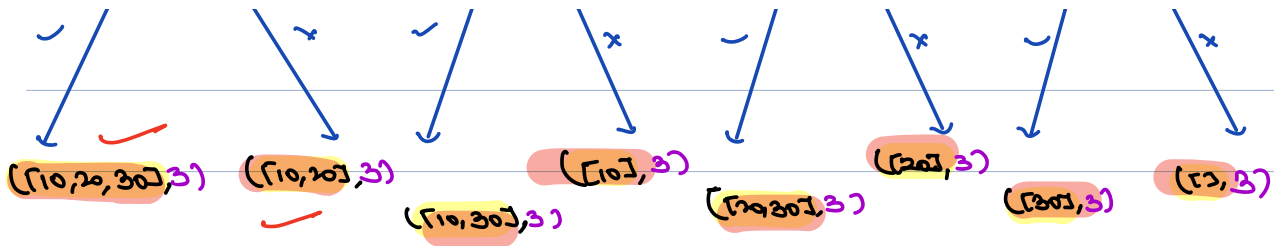
$([10, 20], 2)$

$([10], 2)$

$([20], 2)$

$([ ], 2)$





List < List < Integer > cur;

void subsets ( int[] arr, List<int> curset, int idx ) {

1     if ( idx == arr.length ) {  
2         print ( curset );     cur.set();  
3         return;  
4         cur.add();  
5         cur.remove();  
6         subsets ( arr, curset, idx+1 );  
7         subsets ( arr, curset, idx+1 );  
8     }

// for every element we have two choices.

// choice 1: pick

curset.add ( arr[idx] );

subsets ( arr, curset, idx+1 );

curset.remove ( curset.size()-1 );

// choice 2: don't pick

subsets ( arr, curset, idx+1 );

= 8:27 Am to 8:37 Am Break

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

S.C  $\rightarrow O(n)$ .

Ques) Print / Generate all N digit numbers,  
using 1 & 2.

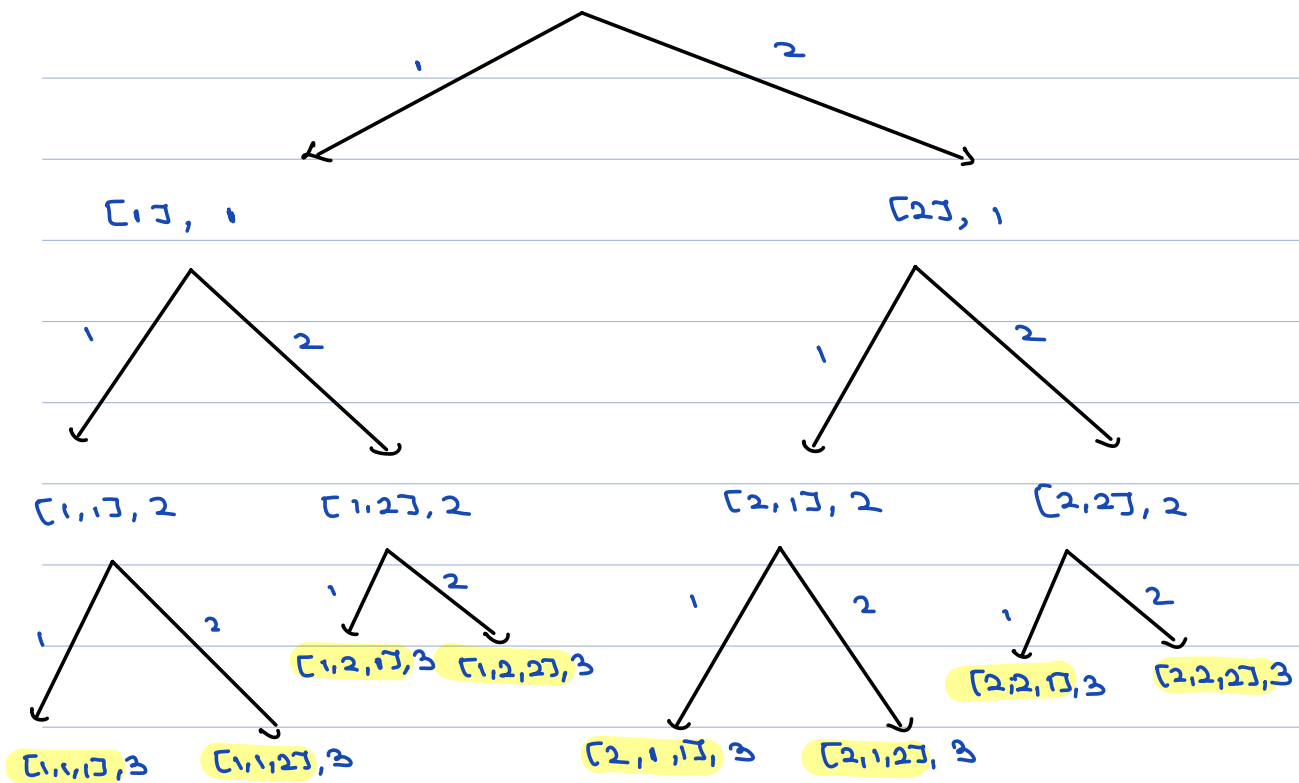
N = 1,  $\begin{matrix} 1 \\ 2 \end{matrix}$

N = 2,  $\begin{matrix} 11 \\ 12 \\ 21 \\ 22 \end{matrix}$

N = 3,  $\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 1 \\ 1 & 2 & 2 \\ 2 & 1 & 1 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \\ 2 & 2 & 2 \end{matrix}$

recursion  
( [ ], 0 )

$\begin{matrix} 1 & 2 & 3 \\ 0 & 1 & 2 \end{matrix}$



```

void subsets ( int[] arr, List<int> curset, int idx) {
1   if (idx == arr.length) {
2       print (curset);
3       return;
4   }
5   // for every element we have two choices.
6   // choice 1: use 1
7   curset.add ( arr[idx] );
8   subsets (arr, curset, idx+1);
9   curset.remove (curset.size()-1);
10  // choice 2: use 2
11  curset.add ( arr[idx] );
12  subsets (arr, curset, idx+1);
13  curset.remove (curset.size()-1);
14 }

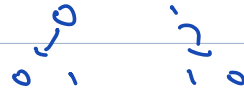
```

$$2^0 + 2^1 + 2^2 + \dots + 2^n$$

$$T.C \rightarrow O(\underline{2^n})$$

$$S.C \rightarrow O(\underline{\infty}),$$

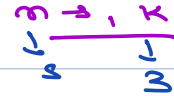
Ques 1) k<sup>th</sup> symbol → Howd



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 0 1 1 0

(n=5, k=2) = 0

(n=4, k=5) = 0

(n=5, k=8) = 1

Obs

1) 0<sup>th</sup> idx is always 0

2) Second half is opp. of first half.

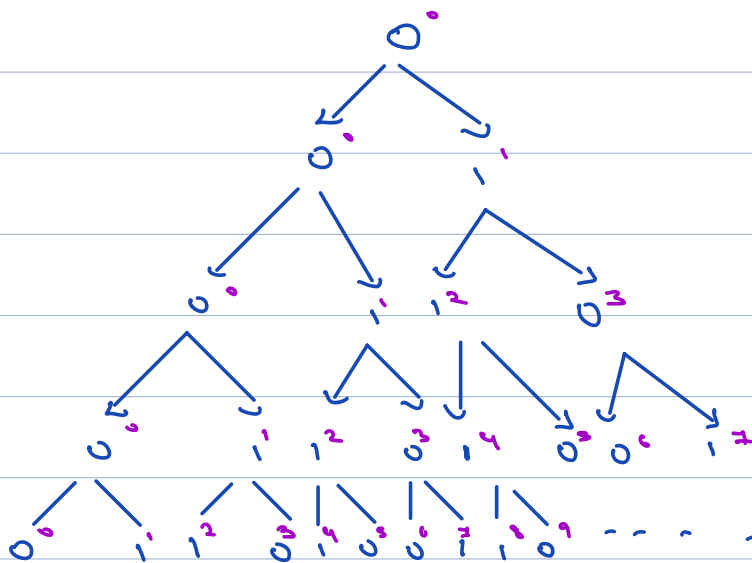
3) Last Idx is alternative.

n=1

n=2

n=3

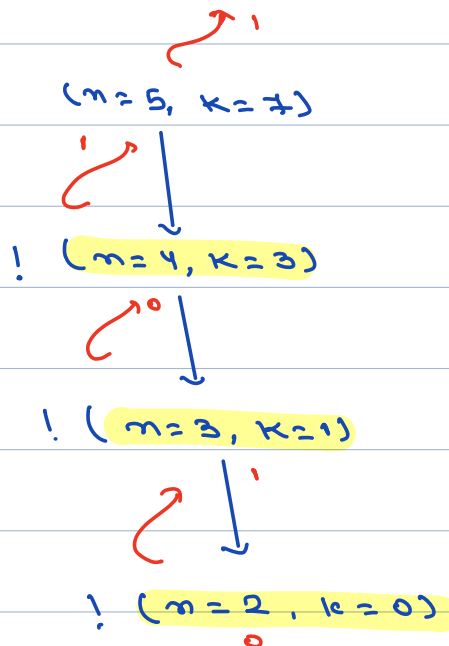
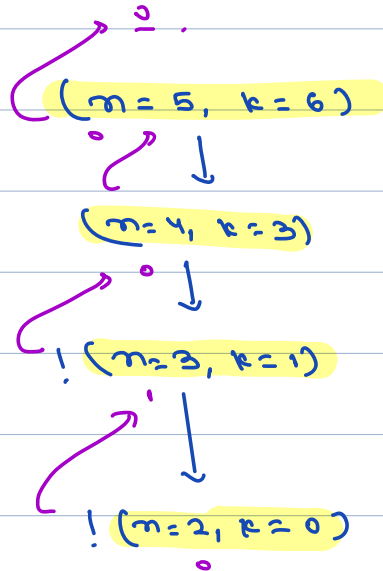
n=4



Obs → 1) Every even idx elem is same as its parent.

2) Every odd idx element is opp. dir  
parent.

3) Parent idx =  $\frac{\text{my idx}}{2}$





```

int kthSymbol ( n, k) {
    if (n==1 || k==0) return 0;
    if (k%2==0) {
        return kthSymbol (n-1, k/2);
    }
    else {
        return 1 - kthSymbol (n-1, k/2);
    }
}

```

T.C  $\rightarrow O(\log k);$

S.C  $\rightarrow \frac{O(\log k)}{(1000)^6}$

$(10)^6$

$n, 2^{n-1}$

$n, \log 2^{n-1}$

$n, n-1, \log 2$

$n \rightarrow$   
 $\frac{n}{2}, \frac{n-1}{2}$   
 $\rightarrow$   
 $\frac{n}{2}, \log 2^{n-1}$

$\frac{n}{2}, \frac{n-1}{2}$

```
void fun ( int idx, list<int> temp) {
```

```
    if (idx == 1) { return; }
```

```
    idx--; temp.add (idx);
```

```
    fun (idx, temp);
```

```
    print (idx);
```

3

fun ( 3 , list );

10k

1, 2

main()

list = 10, 20, 30;

fun ( list, idx );

print (list);

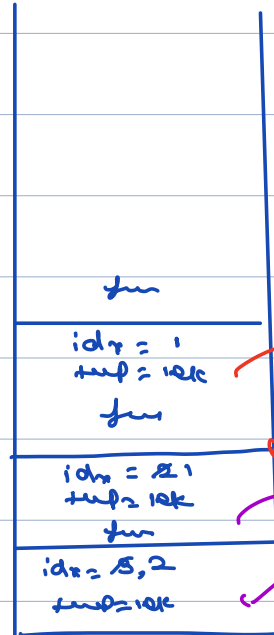
fun ( list, idx ) {

list.add (10);

list = new list<> (1);

idx++;

3



fun

list = 10k  
idx = 0;

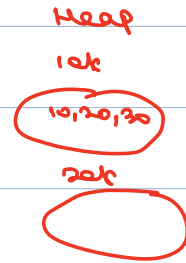
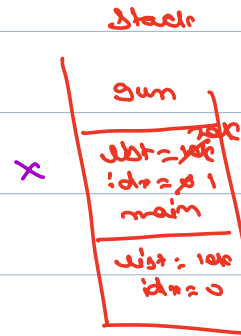
10k

10

```

main() {
    idx = 0;
    list = 10, 20, 30;
    gen ( list, idx );
    print list;
}

```



```

gen ( list, idx ) {

```

```

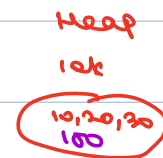
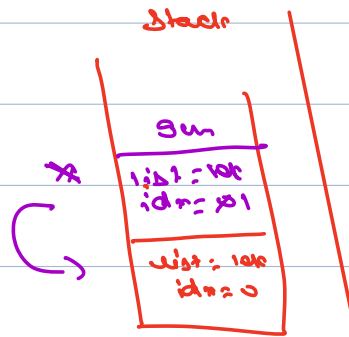
    list = new list <> (1,
        idx++);
}

```

```

main() {
    idx = 0;
    list = 10, 20, 30;
    gen ( list, idx );
    print list;
}

```



```

gen ( list, idx ) {

```

```

    list.add(100);
    idx++;
}

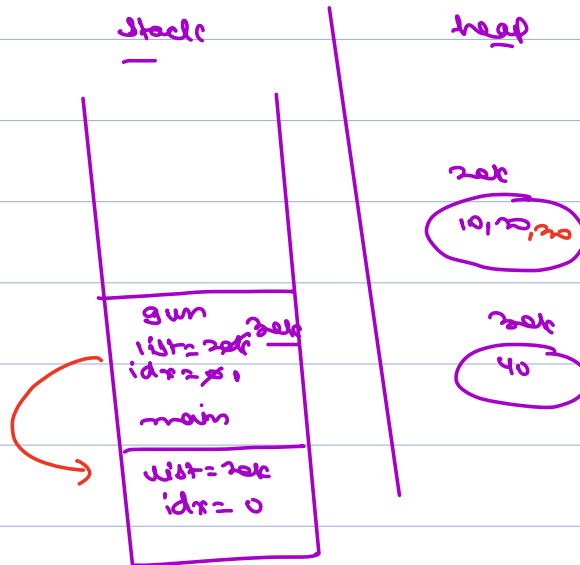
```

```

public static void main(String[] args) {
    // System.out.println("Hello, World!");
    ArrayList<Integer> list = new ArrayList<>();
    list.add(10);
    list.add(20);
    int idx = 0;
    gun(list, idx);
    System.out.println(list + " " + idx);
}

public static void gun(ArrayList<Integer> list, int idx){
    list = new ArrayList<>();
    list.add(40);
    idx++;
}

```



$n = 5$

`print (n--);`

