

Class 77Backtracking

181, 292, 707, 818, 929

$$n=3, k=7$$

181	292	707	818	929
\u \u 7 7				

(k) abs

~~1-9~~

~~1-9~~

abs

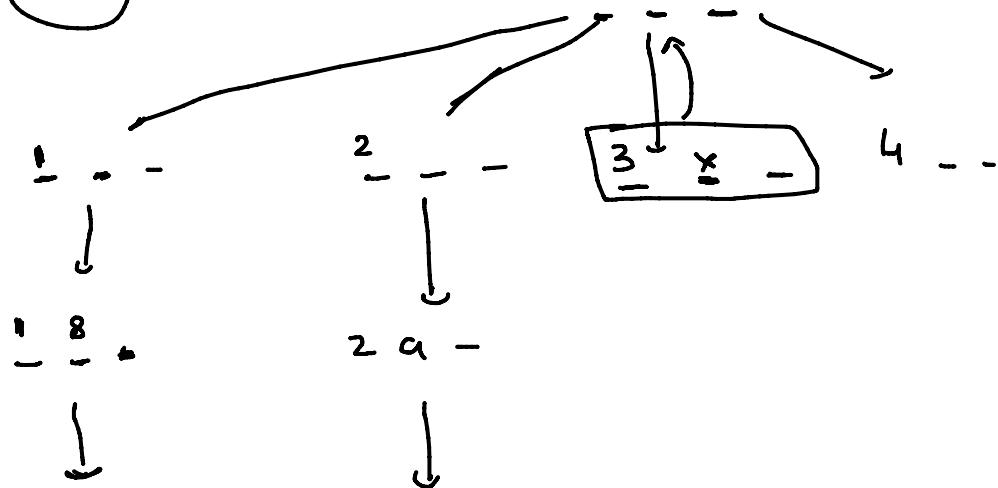
1-9  
k=7

1-9  
(1-9)

k

3+7=10

3-7 < 0

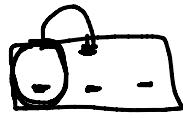


1 8 1  
2 9 2  
(0-9)  
last digit + k  
" k

... compare diff / int n, int k

vector<int> have same cons diff ( int n, int k )

{ string temp = " ";



getnos( 0, n, k, -1, temp );

return finalans;

~

vector<int> finalans;

void getnos( int idu, int n, int k, int lastdigit, String temp )

// Base case

{ if ( idu == n )

! .  
— .

{ int no = stoi( temp );

finalans.push\_back( no );

~ reduce;

if ( idu == 0 )

{ For ( [ i = 1 ] ; c = 9; i++ )

{ temp.push\_back( [ i + '0' ] );

getnos( idu, n, k, [ i ] + temp );

temp.pop\_back();

~

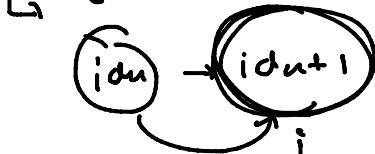
else

{ int digit1 = lastdigit + 1c;

int digit2 = lastdigit - 1c;

'6' → 48

'1' → 49



```

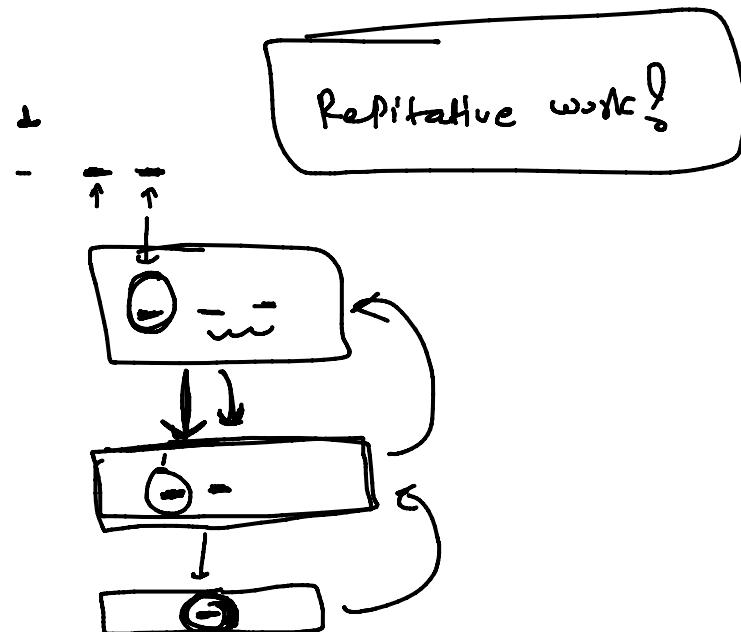
if (digit1 >= 0 and digit1 <= 9)
    temp.push-back (digit1 + '0');

    getnos (idu1, n, lc, digit1, temp);
    temp.pop-back();

    if (digit2 >= 0 and digit2 <= 9)
        temp.push (digit2 + '0');

        getnos (idu1, n, lc, digit2, temp);
        temp.pop-back();
    }
}

```



Non-decreasing

||

1111

Strictly increasing

12345

Combinations

SofA

Sum, I, ST, DS

1, 2, 3, 4, 4

4 4 3 2 1

(1, 2) | (1, 3) | (1, 4) | (1, 2, 3)

4, 4

6  
5  
Size = 2  
  
2 4 7 [6, 3, 7]

✓  
X

4, 6, 3, 7

4, 7, 6, 7  
(x)

map

4, 6, 7, 7

4, 6, 7  
[3] 7

4, 6, 7 [7]

4, 6, 7  
4, 6, 4

vector<vector<int>> findSubsequences(vector<int> &arr)

{  
vector<int> temp;  
↓

```
getdel (nume, 0, temp);
```

reduce arr;

9

vector<vector<int>> arr;

```
void getdel (vector<int> nume, int i, vector<int> temp)
```

{

if (i == n)

```
{ if (temp.size() >= 2)
```

```
    arr.push_back (temp);
```

// reduce;

//

```
    if (temp.size() >= 2) arr.push_back (temp);
```

Set cint > ss;

```
for (int i = 0; i < n; i++)
```

```
{ if (ss.find (nume[i]) != ss.end())
```

continue;

//

else

```
{ if (temp.size() == 0)
```

```
    temp.push_back (nume[i]);
```

```
    getdel (nume, i + 1, temp),
```

```
    temp.pop_back();
```

Duplicated, unique

[4, 3, 3, 2]

[4, 3, 2, 1]

[4, 3, 2]

[4, 3]

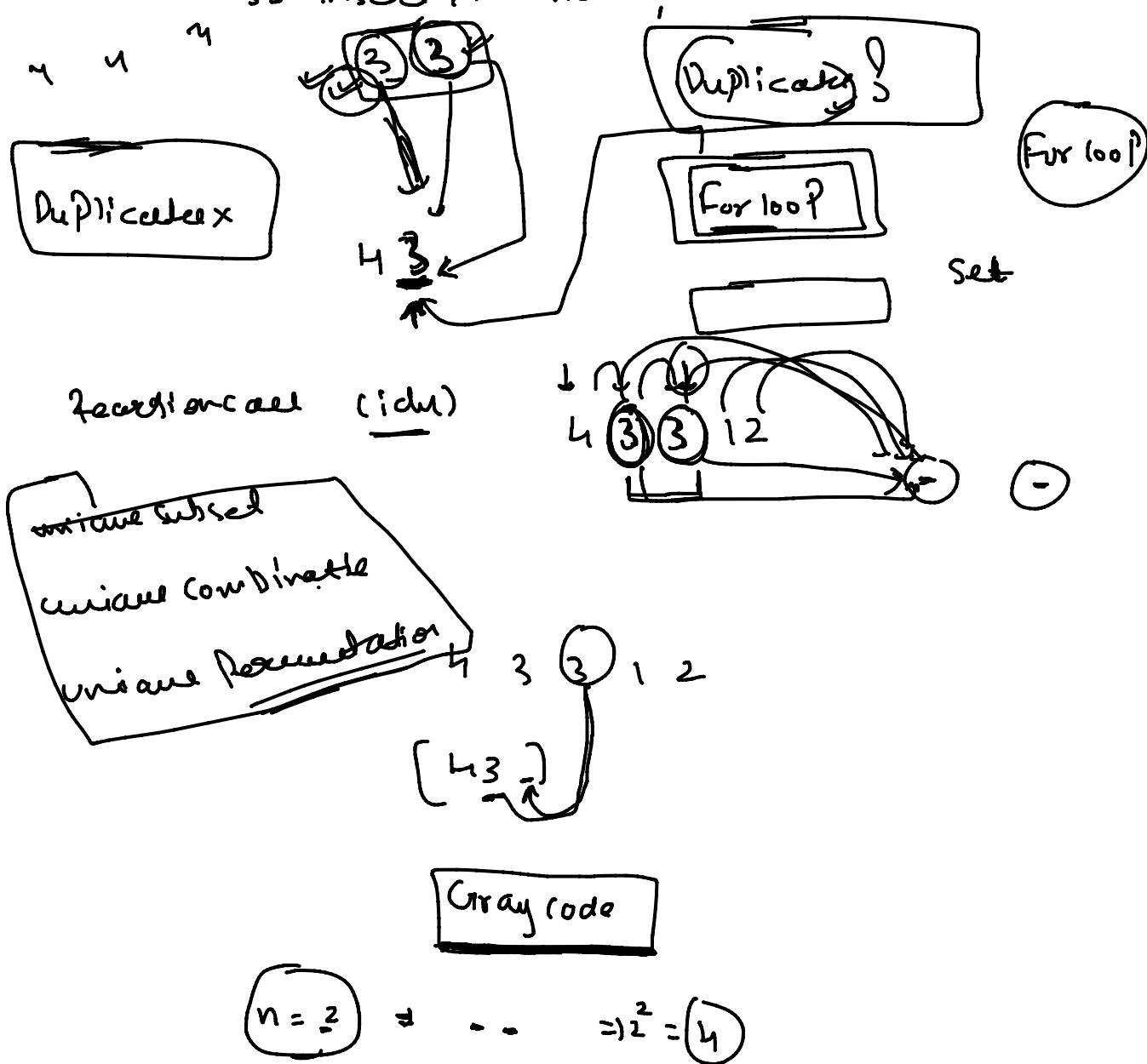
[4, 3, 3]

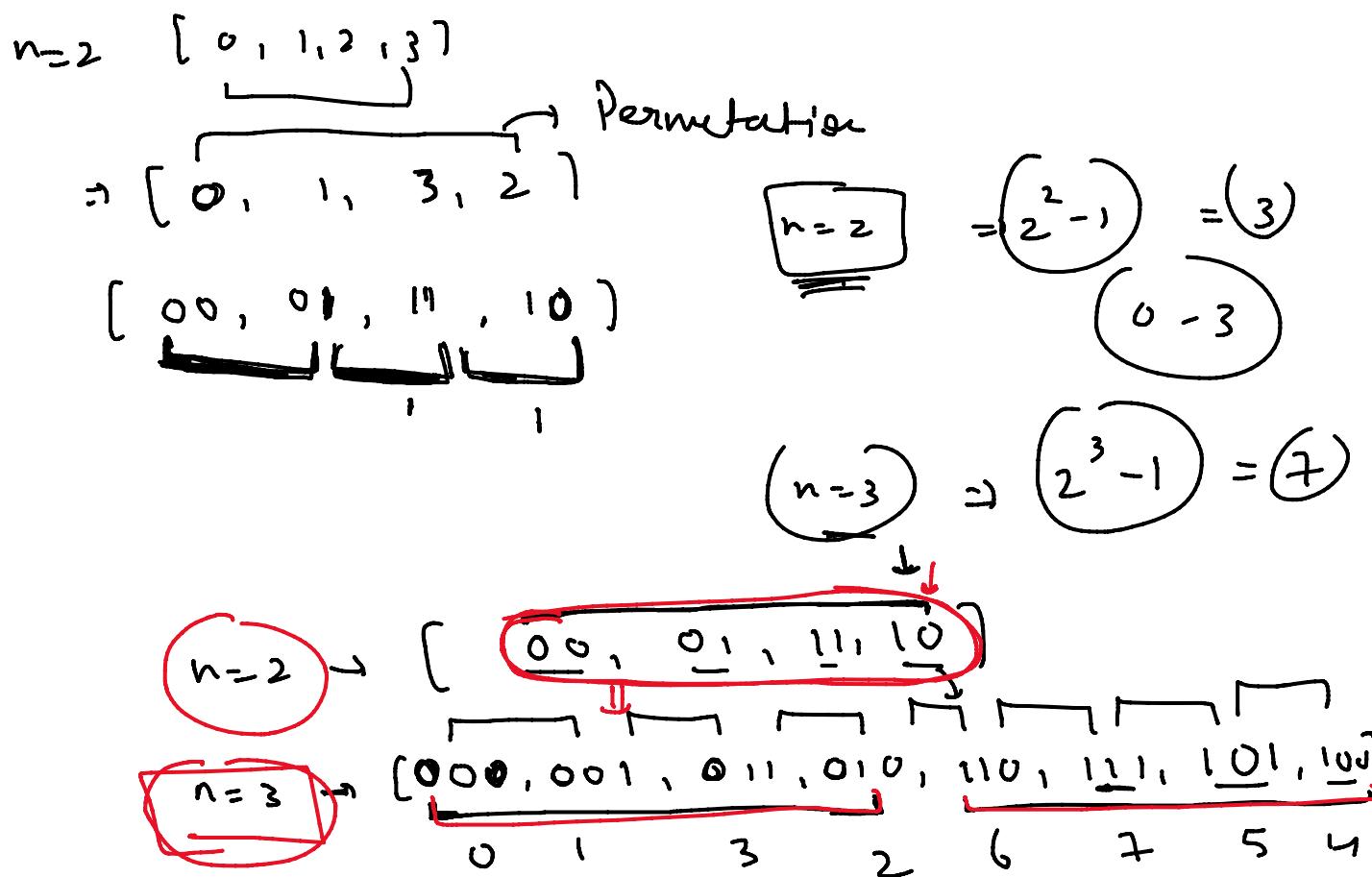
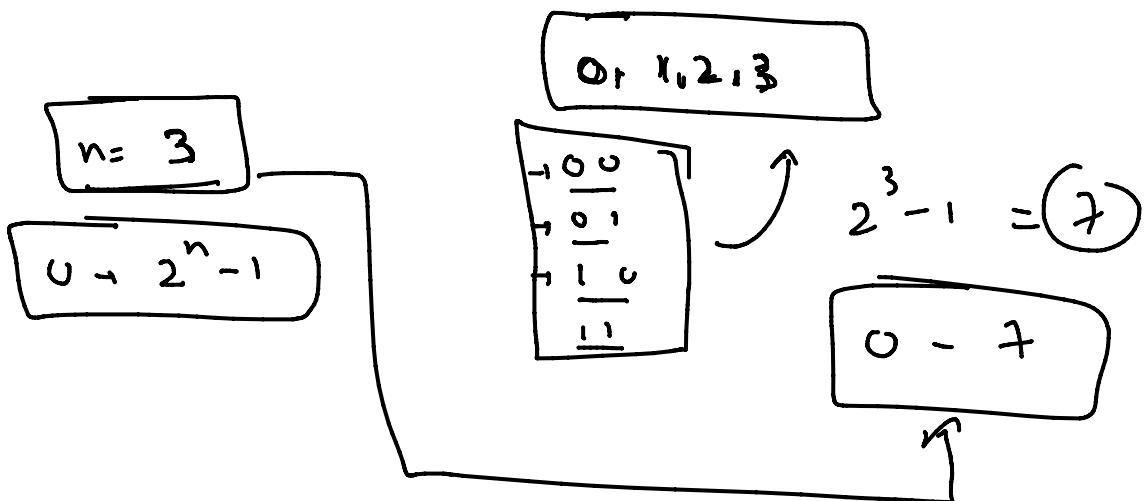
[4, 3, 3, 2]

```

ss.insert (new (idn));
y
else
{
    if (temp.back() <= new(idn))
    {
        temp.push_back (new(idn));
        get all (new, idn + 1, temp);
        temp.pop_back();
        ss.insert (new(idn));
    }
}

```





$n$   
 $n-1$   
 $n-2$   
 $n-3$   
 $2^1 - 1 = 1$   
 $\{0, 1\}$   
 $\{00, 01\}$   
 $if(n=1) \Rightarrow$   
 $vector<int> grayCode (index)$   
 $\& if(n=1)$   
 $grayCode[0, 1] \sim$

...  
 ↓  
 (n-3) 32,600  
 ↓  
 ! 00000  
 100  
 (n=3)  
 00,01,11 10  
 010  
 0 10  
 1 00  
 110

\* If  $n = 1$   
 return  $\{0, 1\}$ ;  
 vector  $\text{int} \rightarrow \text{ans} = \underline{\text{graycode}(n-1)}$ ;  
 For ( $i = \underline{\text{ans.size()}-1}; i \geq 0; i--$ )  
 {  
 ans.push-back ( $\text{ans}[i] \uparrow (1 \ll n-1)$ )  
 }  
 return ans;