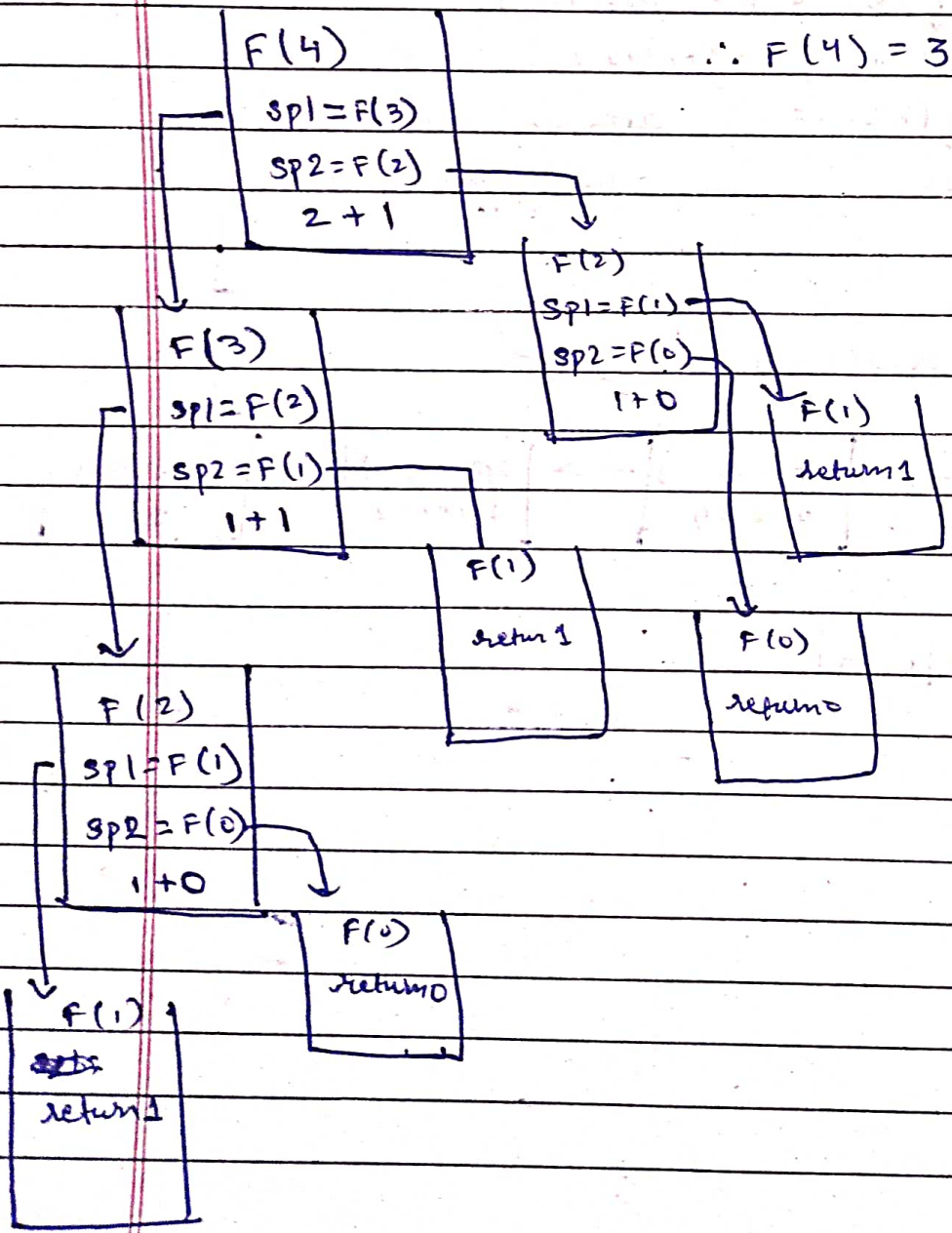
Page No.:
Date : / /

fibonacci (similar to preorder traversal)

BP = F(i)

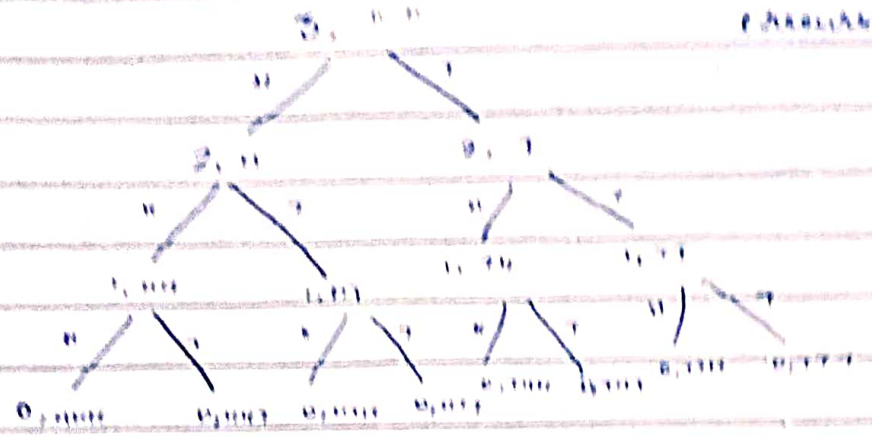
SP = F(i-1) & F(i-2)

```
F(int i)
{
    if(i<=1)  return i
    int sp1 = f(i-1);
    int sp2 = f(i-2);
    return sp1 + sp2
}
```



∴ F(4) = 3

n coins (similar to multiverse)



```
gp = cf(n,path)        sp = cf(n-1, path)

cf (int n, String path)
{
    if(n==0) sop(n(path)
             return;
    cf (n-1, path+"H")
    cf (n-1, path+"T")
}
```

Subsequence of String

what is substring ⊆ contiguous sequence, subset

subset jo sequence me hai & may or may not be contiguous = subsequence (2^n)

""	a	ab	abc	abcdf
	b	ac	abd	
	c	ad	acdf	
	d	bc	bcd	
		bd		
		cd		
$^4C_0$	$^4C_1$	$^4C_2$	$^4C_3$	$^4C_4$

Page No.:
Date : / /

Subsequence is combination in sequence

order matter = permutation

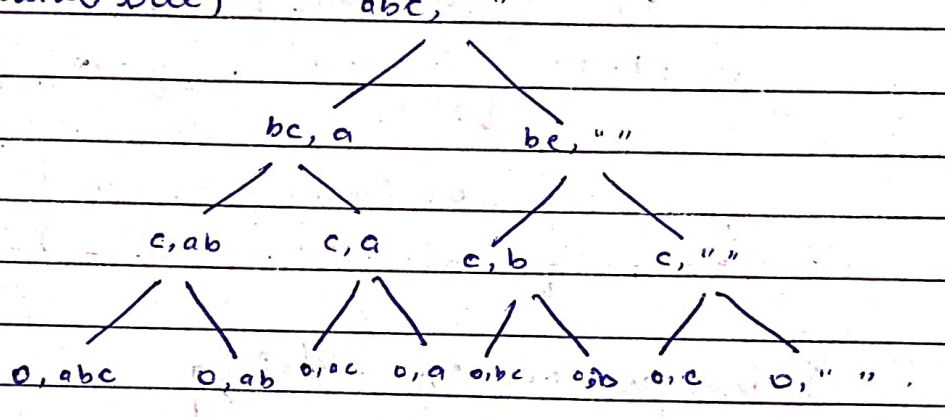
order no matter = combination

${}^nC_r$ = no of ways to select r things out of n things in which order doesn't matter

But

(team बनाना) Subsequence = ways to select r things out of n things which are in sequence but we select only one which is in sequence.

(recursive tree)



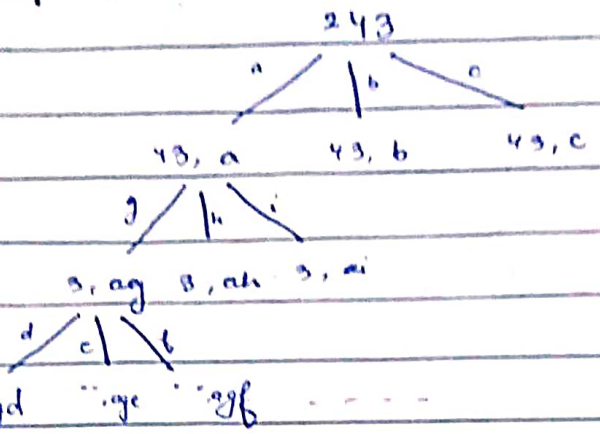

```
subseq (String str, String team)
{
    if (str.isEmpty()) SOPln(team)
                       return
    char ch = str.charAt(0)
    String remain = str.substring(1)
    subseq (remain, team+ch)
    subseq (remain, team)
}
```

Page No.:
Date: / /

Letter combination of a phone number

BP : n

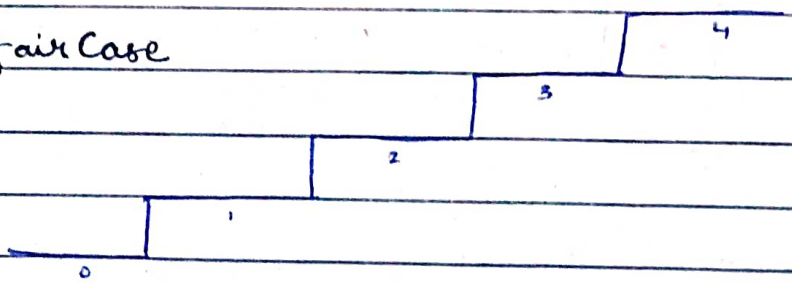
SP = n-1



to pass info in a f" frame < global variable / pass as argument

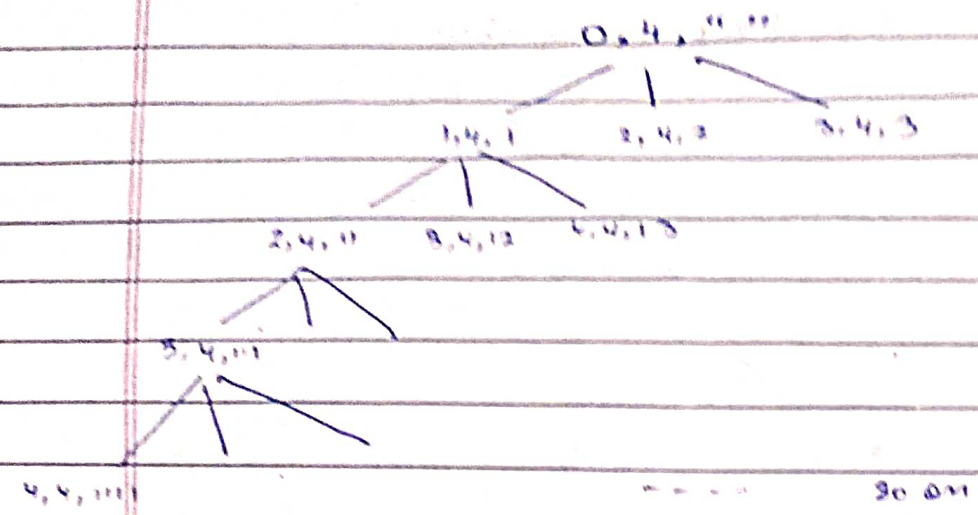
```
static String[] optionsall = {"", "", "abc", "def", "ghi", "jkl", "mno",
                              "pqrs", "tuv", "wxyz"};

solve (String digits, String ans)
{
    if (digits.isEmpty()) SOPln(ans) return;
    char button = digits.charAt(0);
    String options = optionsall[button - '0'];
    String remains = digits.substring(1);
    for (int i=0; i<options.length(); i++)
        solve (remains, ans+option.charAt(i));
}
```

Staircase



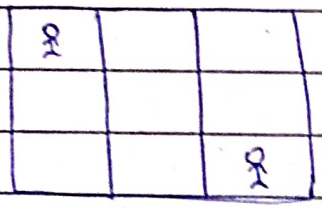
BP : 0-4

SP : 1-4




```
Solve (int curr, int dest, String path)
{
    if (curr == dest) sopln(path) return
    if (curr > dest) return
    Solve (curr+1, dest, path+"1")
    Solve (curr+2, dest, path+"2")
    Solve (curr+3, dest, path+"3")
}
```

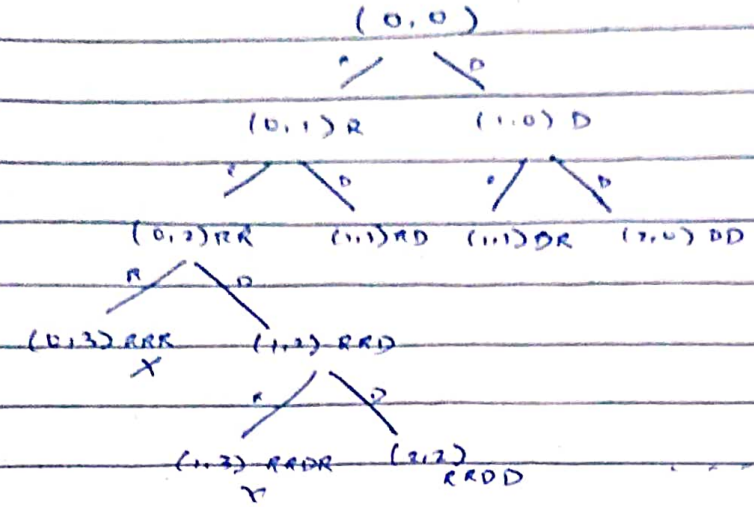
Maze path



RRDD
RDRD
RDDR
DRRD
DRRR
DRDR

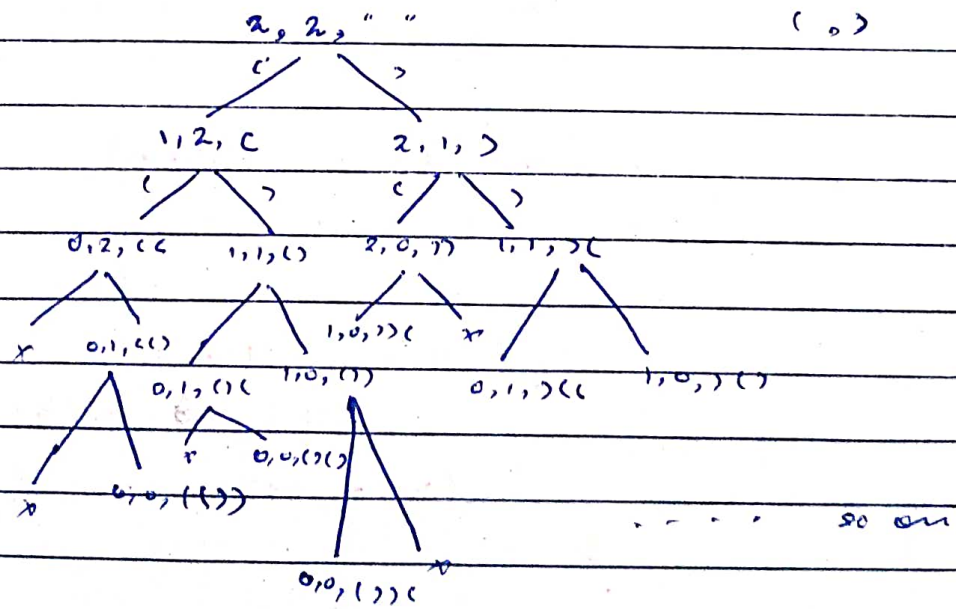
BP = 0, 2, 2, " "

SP = 1, 1, 2, " " / 1, 2, 1, " "



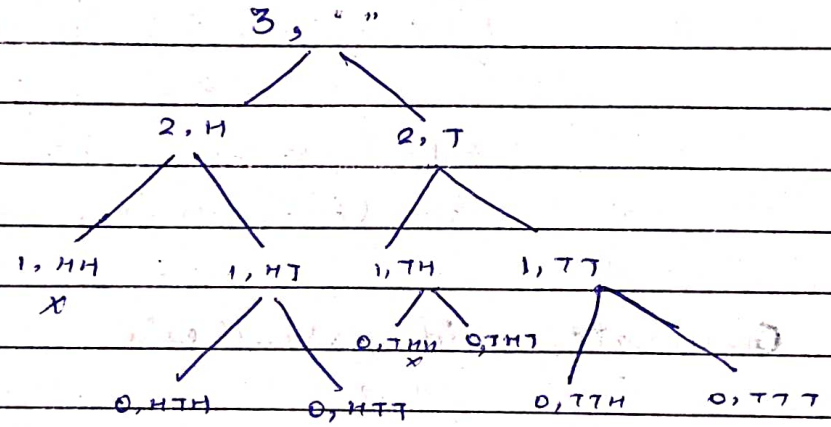

```
Solve ( curr_r , curr_c , dest_r , dest_c , path )
{
    if (dest_r == curr_r && dest_c == curr_c)
        SOPln(path) return
    if (curr_r > dest_r || curr_c > dest_c)
        return
    Solve (curr_r + 1, curr_c , dest_r , dest_c , path + "R")
    Solve (curr , curr_c + 1, dest_r, dest_c , path + "D")
}
```

Generate valid paranthesis



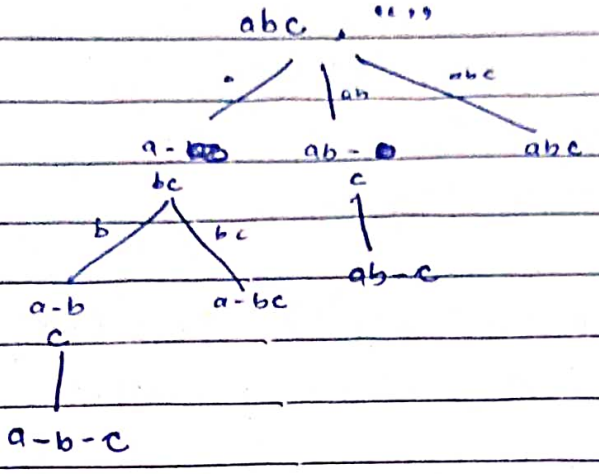
```
solve (op, cl, path)
{
    if (op == 0 && cl == 0) sopln(path) return;
    if (op > cl)            return
    if (op > 0)
    {
        solve (op-1, cl, path + "(")
    }
    if (cl > 0)
    {
        solve (op, cl-1, path + ")")
    }
}
```

Coin toss such that no two heads are consecutive.




```
CT (n, path, isHead)
{
    if (n == 0)   sopln (path) return
    if (isHead == false)
        CT (n-1, path + "H", true)
    CT (n-1, path + "T", false)
}
```

partitioning of a String




```
solve ( table , bag )
{
    if ( table.IsEmpty() ) SOPln (bag)
    for (int i=1 ; i<= table.length() ; i++)
    {
        piece = table.substring (0,i);
        remain = table.substring (i)
        solve ( remain , bag +"-" + piece )
    }
}
```

palindromic partitioning of a String

Steps to build intutions

1. tukde tukde krna
2. tasty tukde krna (tasty == palindromic)
3. Solve it with ArrayList (kuch to gadbad hai daya)
4. make arraylist behave like String
   (matlab) Strings are immutable har baar nya String bnata hai. Waise hi arraylist ko behave krwao)

Page No.:
Date : / /

5. Now solve it with ArrayList of ArrayList.

(1) tasty tukda with String

```
Solve (table, bag)
{
    if (table.isEmpty()) sopln(bag)
    for (int i=1; i<=table.length(); i++)
    {
        piece = table.substring(0,i)
        if (isPalin(piece))
        {
            remain = table.substring(i)
            solve(remain, bag+"-"+piece)
        }
    }
}

isPalin(piece)
{
    s=0   e=piece.length()-1;
    while(s<e)
    {
        if(piece.charAt(s) != piece.charAt(e))
            return false;
        s++  e--
    }
    return true;
}
```

(2) Solve it with arrayList (tasty tukda) (tukda)

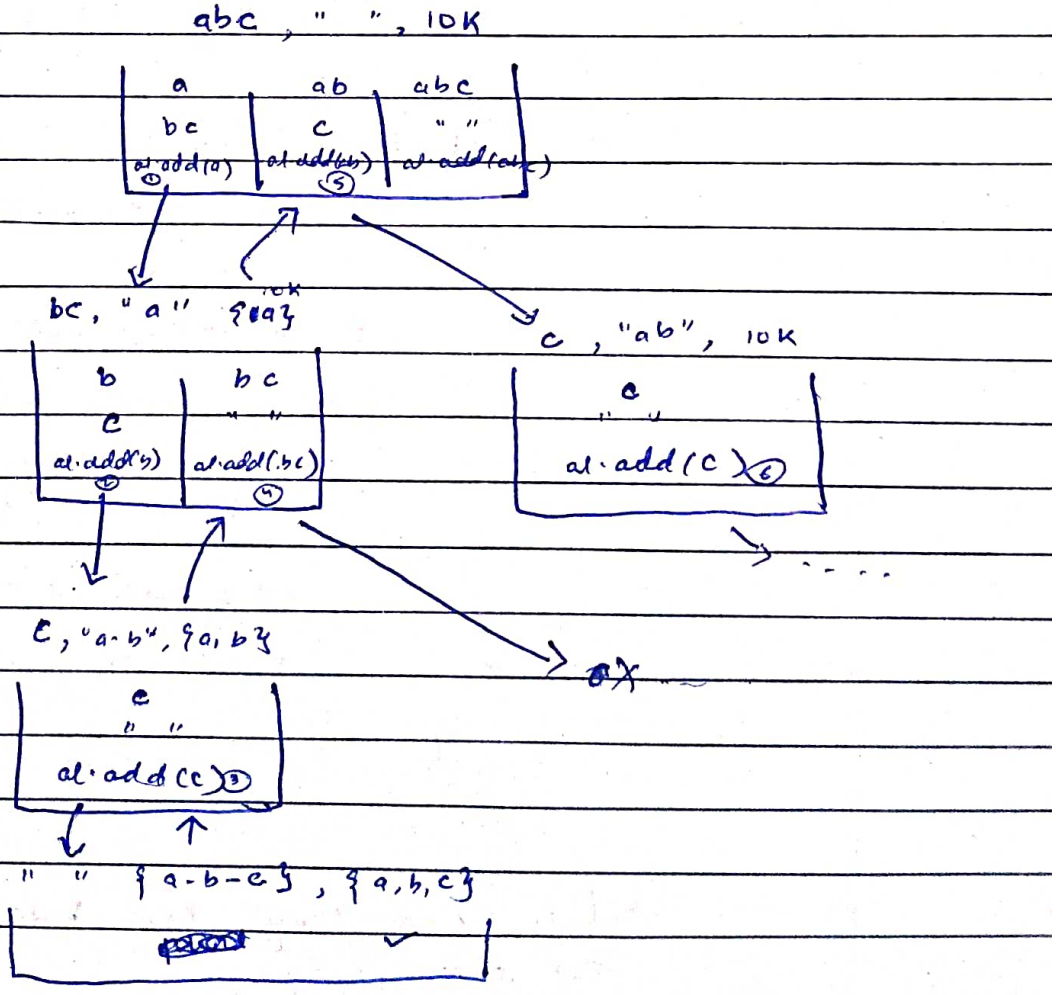
```
solve (table, bag, al)
{
    if (table.isEmpty) return;
    for (int i=1; i<=table.length(); i++)
        p = table.substring(0,i)
```

```
solve (table, bag, al)
{
    if (table.isEmpty()) sopln(al)
    for (int i = 1; i <= table.length(); i++)
    {
        piece = table.substring(0, i)
        remain = table.substring(i)
        al.add(piece)
        solve(remain, bag + "-" + piece, al)
    }
}
```

aapko lga aapne bahoot hi badhiya kr diya pr aap poorn roop se galat ho

Let's dry run where we are wrong

al = { a, b, c, bc, ab, c, abc ... }



Page No.:
Date: / /

Finally, ab samjh aaya kaha galat thi tum kahna nhi?

Let's understand

jitne iteration ho rhe h arraylist sbko apne list me add kr rha h chahe vo iteration tak da ho ya nhi ho

non-primitive data type se address pass hota ha.

(3) Aao ehalo galti sudhare
Arraylist ko string ki tarah behave krwaye

```
solve(table, bag, al)
{
    if(table.isEmpty()) sOPen(al)
    for(int i=1; i<=table.length(); i++)
    {
        piece = table.substring(0,i)
        remain = table.substring(i)
        Arraylist<String> copy = new
                                 Arraylist<>(al);
        copy.add(piece)
        solve(remain, bag+"-"+piece, copy)
    }
}
```

Shit itna copy kon krta ha, exam me UFM lag jaega.

Page No.:
Date: / /

(4) Same arraylist me kro

```
solve (table, bag, al)
{
    if (table.isEmpty()) sopln(al)
    for (int i=1; i<= table.length(); i++)
    {
        piece = table.substring(0,i)
        remain = table.substring(i)
        al.add(piece)
        solve(remain, bag + "-" + piece, al)
        al.remove(al.size()-1);
    }
}
```

(5) Arraylist of Arraylist (leetcode me aisa hi hota hai)

```
solve (table, bag, al, biggy)
{
    if (table.isEmpty())
    {
        biggy.add(new ArrayList<>(al));
    }
    for (int i=1; i<= table.length(); i++)
    {
        piece = table.substring(0,i);
        if(isPalin(piece))
        {
            remain = table.substring(i)
            al.add(piece);
            solve(remain, bag+"-"+piece, al, biggy)
            al.remove(al.size()-1);
        }
    }
}
```

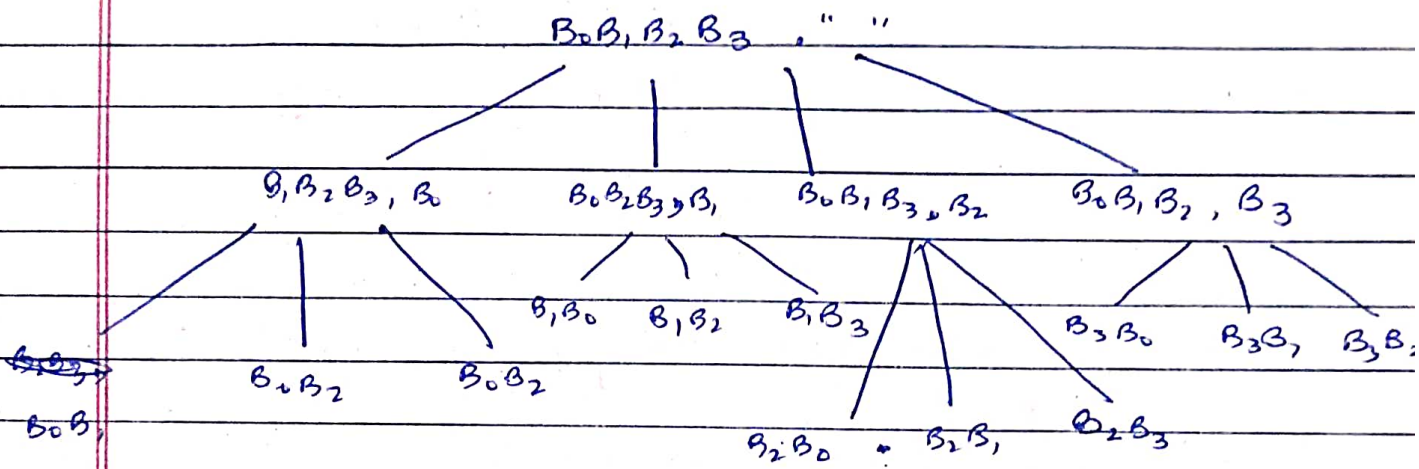
Page No.:
Date: / /

permutation of boxes

B_0	B_1	B_2	B_3

give all arrangements of one box at a time.

$B_0 B_1$	$B_1 B_0$	$B_2 B_0$	$B_3 B_0$
$B_0 B_2$	$B_1 B_2$	$B_2 B_1$	$B_3 B_1$
$B_0 B_3$	$B_1 B_3$	$B_2 B_3$	$B_3 B_2$



if you get a chance to pick a dabba as many times as you like then

```
solve ( n, r, path )
{
    if (r==0) soPln (path) return
    for (i=0; i<n; i++)
        solve (n, r-1, path+"B"+i);
}
```

agr ye shakti tmse cheen li jaaye ~~kisi~~ aur ab tap ek dabba ko ek baar hi pick kr skte ho

```
main()
{
  static boolean isVisited[] = new boolean[n]
   Solve (n, 2, " ", isVisited[])
}

solve (n, s, path, isVisited)
{
    if (s==0) sopln (path); return;
    for (i=0; i<n; i++)
     {
        if (isVisited[i] == false)
        {
            isVisited[i] = true;
            Solve (n, s-1, path + "B" + i, isVisited)
        }
    }
}
```

ye kaam krna chahiye
OOPs
this won't work

chalo smjhte hai kyu

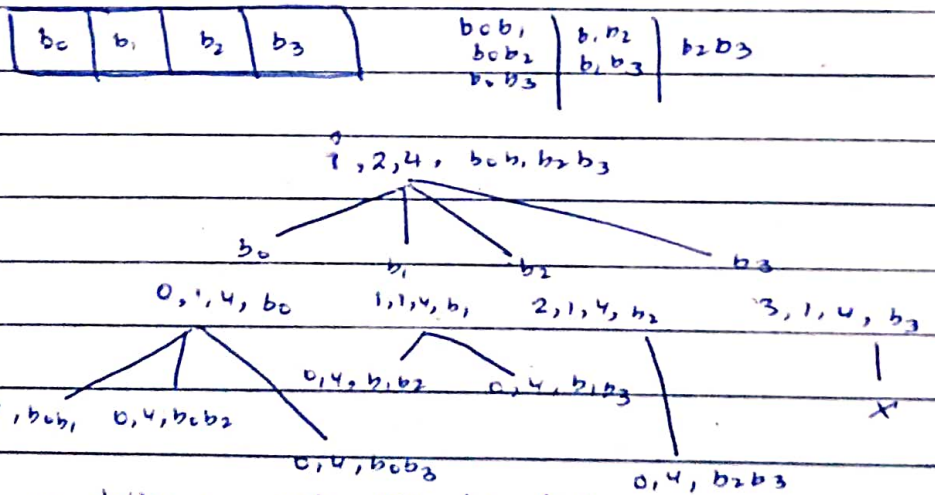
cuz ye ek saath do boxes k isVisited ko true kr diya like
jb B_0 pe gya to B_0 & B_1 dono ko true kr diya aur B_0B_2 pe chla gya aur exit ho gya

issliye ab hm tshirt pehen kr utarenge bhi

```
Solve (n, r, path, isVisited)
{
    if (r==0) soPln(path) return
    for (i=0; i<n; i++)
    {
        if (isVisited[i] == false)
        {
            isVisited[i] = true
            Solve (n, r-1, path+"B"+i, isVisited)
            isVisited[i] = false;
        }
    }
}
```

Combination of boxes ${}^4C_2$

combination = team bannana permutation = arrangement matters



combition not PT teacher

```
(1)  comb (n, r, prev, path)
     {
         if (r==0) soPln(path) return;
         for (i = prev+1; i<=n; i++)
             comb (n, r-1, i, path+'b'+i)
     }
```

Page No.:
Date: / /

② combination wit Student

```
combU (n, curr, r, path)
{
    if (curr == n) return
    if (r == 0) SOPln(path) return
inclusive combU(n, curr+1, r-1, path + 'b' + curr)
exclusive combU(n, curr+1, r, path)
}
```

+ve base case = 2 bnde aa gye

-ve base case = String ki khatam ho gye

PT teacher & student wli story bhul gyi