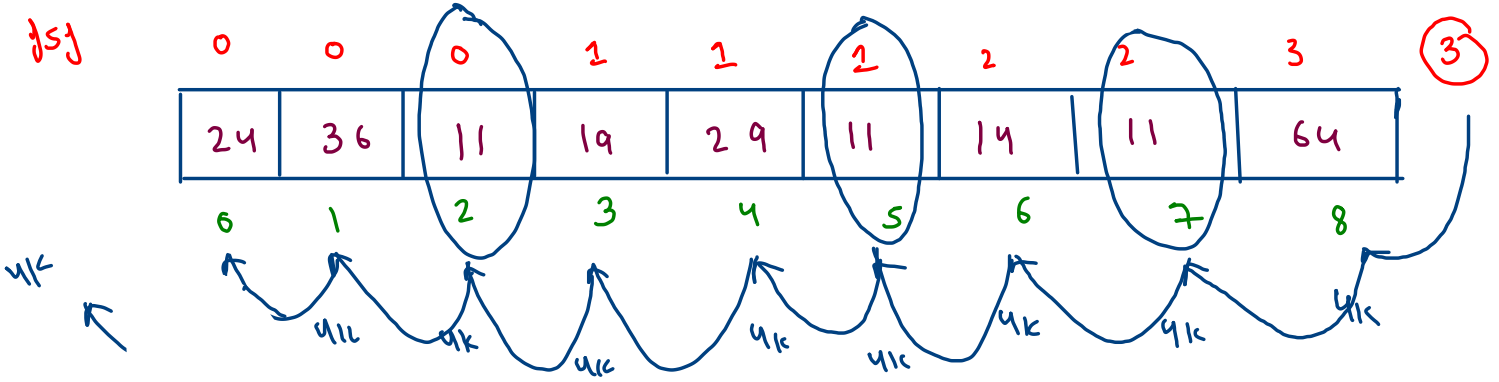
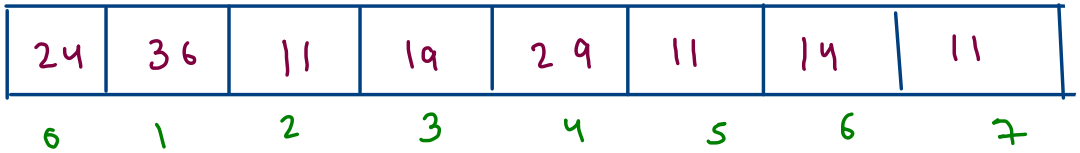
all indices



	5	7

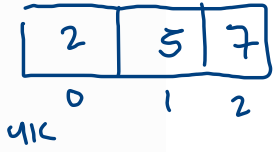
yk

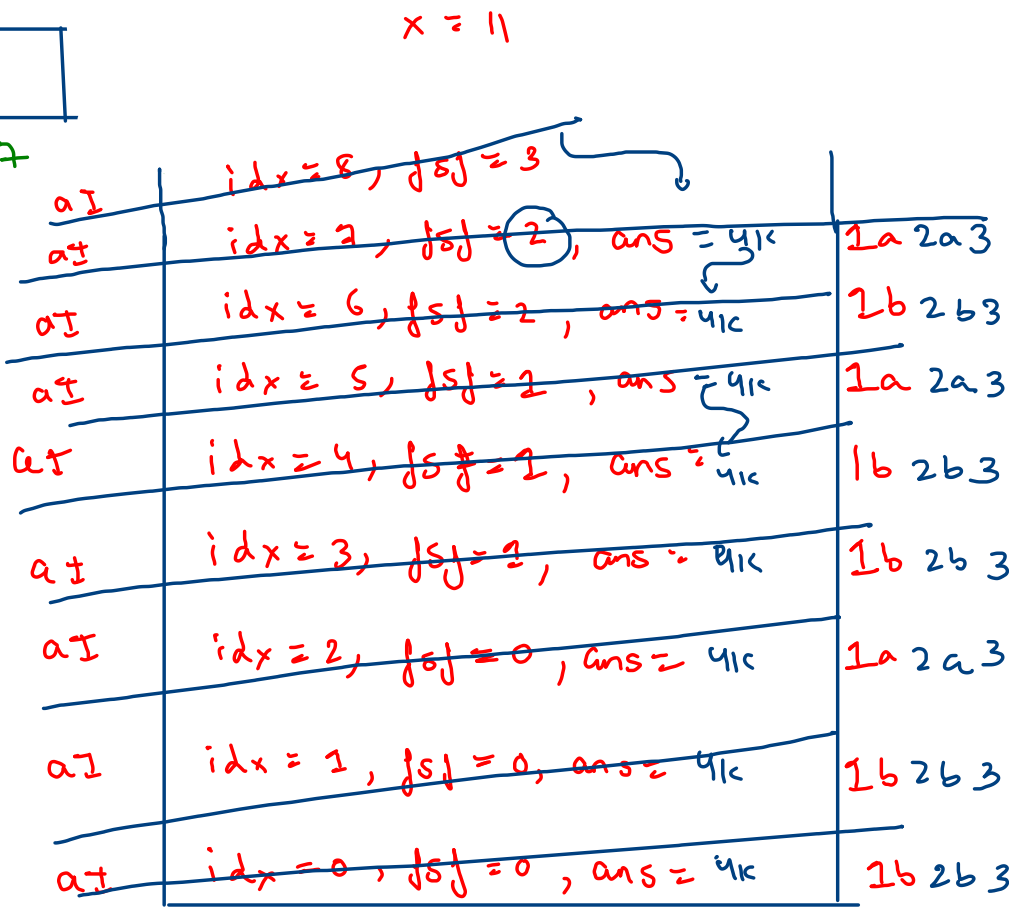
val = 11




```java
public static int[] allIndices(int[] arr, int x, int idx, int fsf) {
    if(idx == arr.length) {
        return new int[fsf];
    }

    int[]ans;
    if(arr[idx] == x) {
        ans = allIndices(arr,x,idx+1,fsf+1);
    }
    else {
        ans = allIndices(arr,x,idx+1,fsf);
    }

    if(arr[idx] == x) {
        ans[fsf] = idx;
    }
    else {
        //do nothing
    }

    return ans;
}
```

1 a b
2 a b
3





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

abc

[, c, b, bc, a, ac, ab, abc]

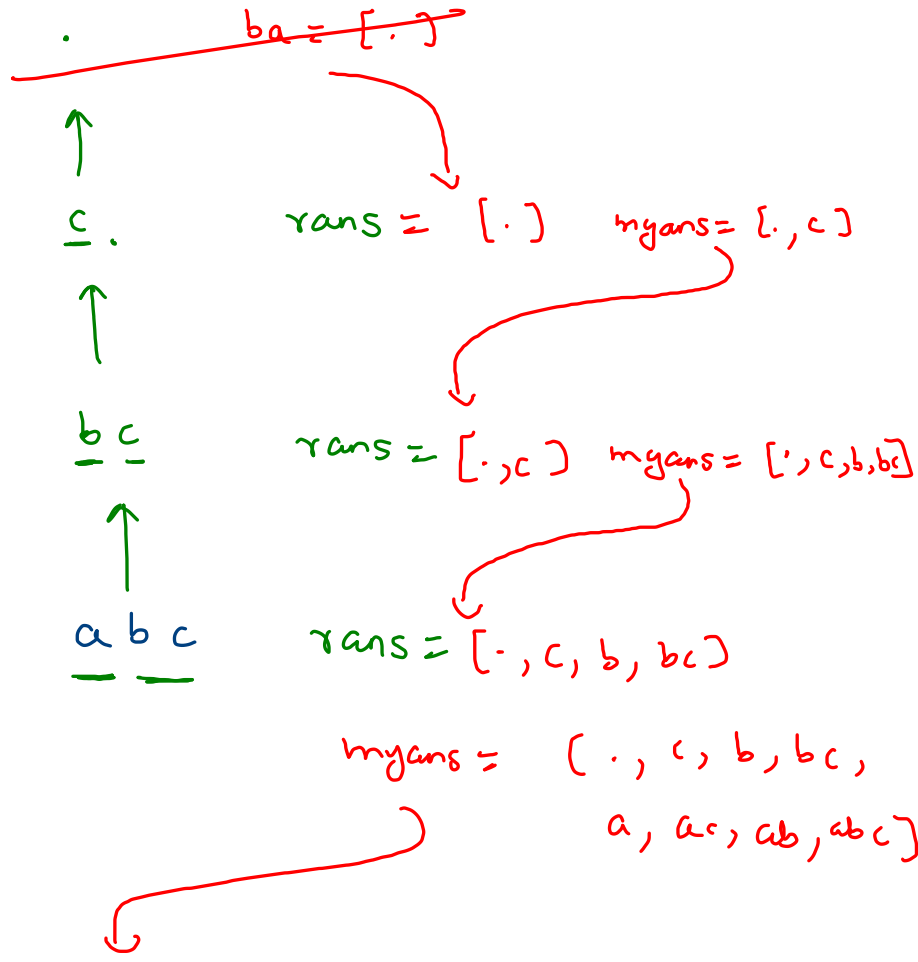
b c
- -
- c
b -
b c

[, c , b , bc)

bc

abc : [, c, b, bc, a, ac, ab, abc)

```java
public static ArrayList<String> gss(String str) {
    if(str.length() == 0) {
        //empty subseq
        ArrayList<String>ba = new ArrayList<>();
        ba.add("");
        return ba;
    }


    char ch = str.charAt(0);
    String ros = str.substring(1);

    ArrayList<String>rans = gss(ros);
    ArrayList<String>myans = new ArrayList<>();

    //ch -> no choice
    for(int i=0; i < rans.size();i++) {
        myans.add(rans.get(i));
    }

    //ch -> yes choice
    for(int i=0; i < rans.size();i++) {
        myans.add(ch + rans.get(i));
    }

    return myans;
}
```

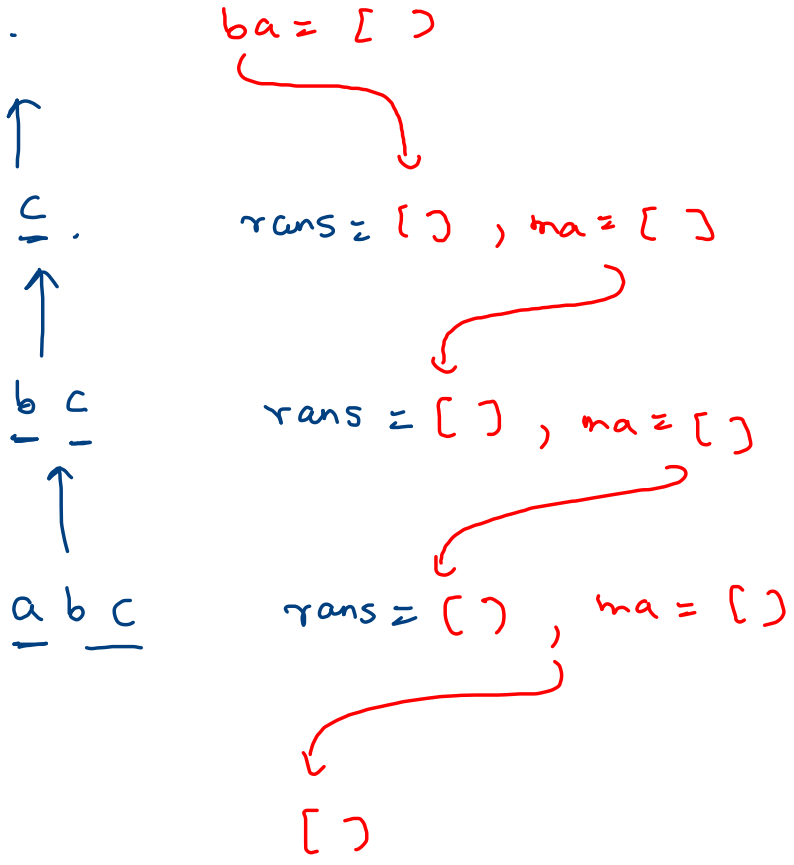


[ ] -> list.size() = 0

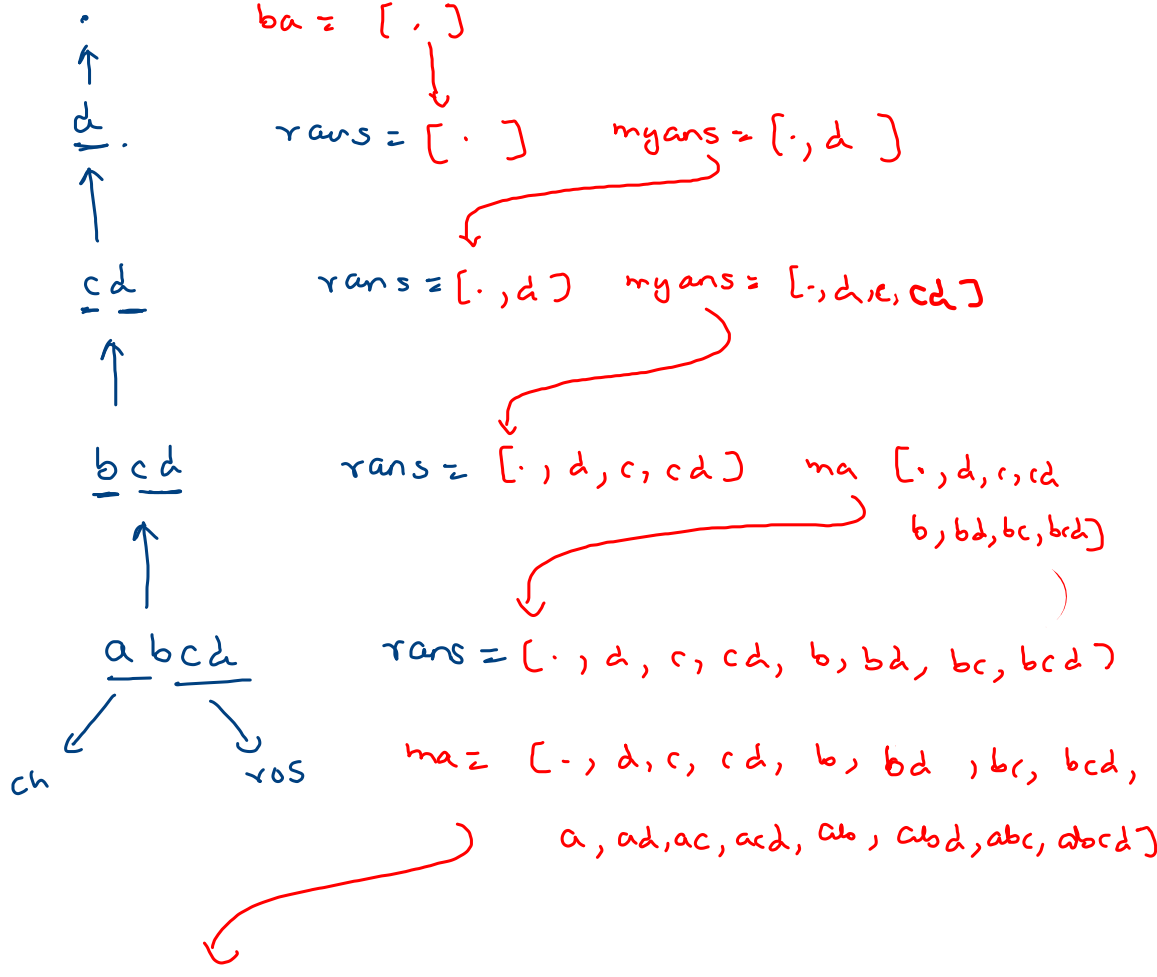
[""] -> list.size() = 1

list.get(0).length() = 0

```java
public static ArrayList<String> gss(String str) {
    if(str.length() == 0) {
        //empty subseq
        ArrayList<String>ba = new ArrayList<>();
        //ba.add("");
        return ba;
    }


    char ch = str.charAt(0);
    String ros = str.substring(1);

    ArrayList<String>rans = gss(ros);
    ArrayList<String>myans = new ArrayList<>();

    //ch -> no choice
    for(int i=0; i < rans.size();i++) {
        myans.add(rans.get(i));
    }

    //ch -> yes choice
    for(int i=0; i < rans.size();i++) {
        myans.add(ch + rans.get(i));
    }

    return myans;
}
```



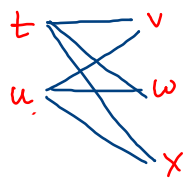
```java
public static ArrayList<String> gss(String str) {
    if(str.length() == 0) {
        //empty subseq
        ArrayList<String>ba = new ArrayList<>();
        ba.add("");
        return ba;
    }


    char ch = str.charAt(0);
    String ros = str.substring(1);

    ArrayList<String>rans = gss(ros);
    ArrayList<String>myans = new ArrayList<>();

    //ch -> no choice
    for(int i=0; i < rans.size();i++) {
        myans.add(rans.get(i));
    }

    //ch -> yes choice
    for(int i=0; i < rans.size();i++) {
        myans.add(ch + rans.get(i));
    }

    return myans;
}
```



0 -> .;
1 -> abc
2 -> def
3 -> ghi
4 -> jkl
5 -> mno
6 -> pqrs
7 -> tu
8 -> vwx
9 -> yz

```
static String[]codes = {".;", "abc","def","ghi","jkl","mno","pqrs","tu","vwx","yz"};
```

7 8



tv, tw, tx, uv, uw, ux

5 6 7

m p t

n q u

o r s

mpt	npt	opt
mpu	npu	opu
mqt	nqt	oqt
mqu	nqu	oqu
mrt	nrt	ort
mru	nru	oru
mst	nst	ost
msu	nsu	osu

67 -> [pt, pu, qt, qu, rt, ru, st, su]

567 -> [mpt, mpu, mqt, mqu, mrt, mru, mst, msu,
npt, npu, nqt, nqu, nrt, nru, nst, nsu,
opt, opu, oqt, oqu, ort, oru, ost, osu]

'm' + 67

'n' + 67

'o' + 67

```
static String[]codes = {".;", "abc","def","ghi","jkl","mno","pqrs","tu","vwx","yz"};
String mycode = codes[ch]; //"mno"
```

Ascii -> american standard code for information interchange

(char to integer)

'a' - 'z' -> 97 - 122

'A' - 'Z' -> 65 - 91

'0' - '9' -> 48 - 57

'8' = '0' + 8

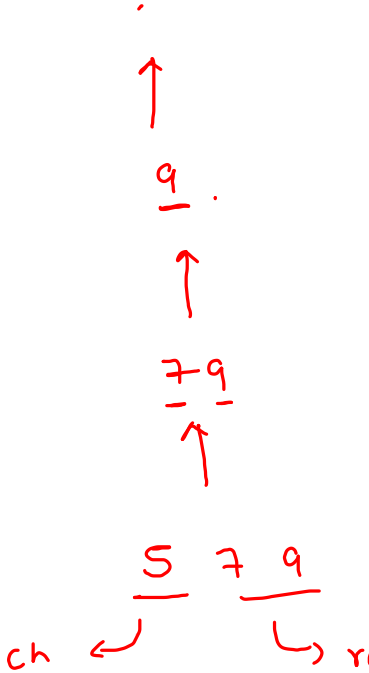
'8' - '0' = 8

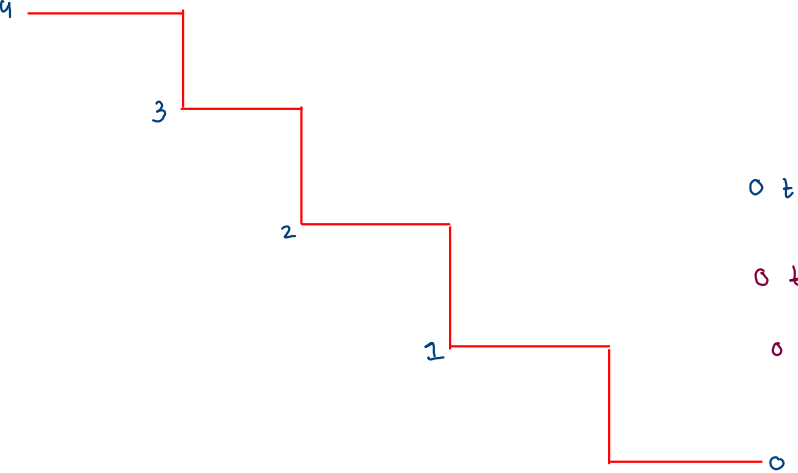
0 -> .;
1 -> abc
2 -> def
3 -> ghi
4 -> jkl
5 -> mno
6 -> pqrs
7 -> tu
8 -> vwx
9 -> yz

```
public static ArrayList<String> getKPC(String str) {
    if(str.length() == 0) {
        ArrayList<String>ba = new ArrayList<>();
        ba.add("");
        return ba;
    }

    char ch = str.charAt(0);
    String ros = str.substring(1);

    ArrayList<String>rans = getKPC(ros);
    ArrayList<String>myans = new ArrayList<>();

    String mycode = codes[ch - '0'];

    for(int i=0; i < mycode.length();i++) {

        for(int j=0; j < rans.size();j++) {
            String code =  mycode.charAt(i) + rans.get(j);
            myans.add(code);
        }
    }

    return myans;
}
```

5 7 9
m t y
n u z
o

total -> 12





1, 2, 3

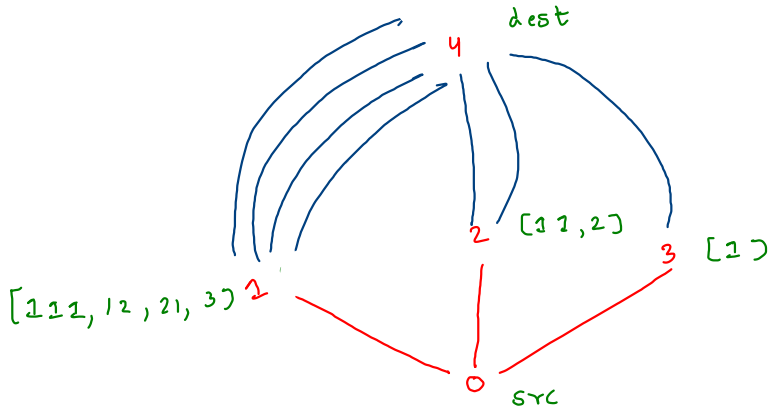
0 to 3 -> [111, 12, 21, 3]

0 to 2 -> [11, 2]

0 to 1 -> [1]

0 to 4 -> 1[0 to 3] + 2[0 to 2] + 3[0 to 1]

0 to 4 -> 1111, 112, 121, 13, 211, 22, 31

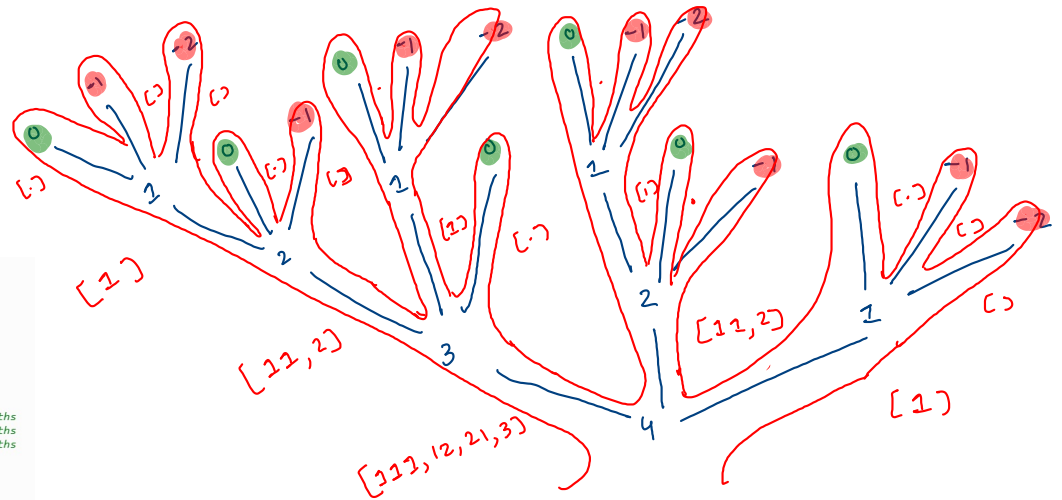


0 to 4 -> 0 to 1 + [1-4]

0 to 4 -> 0 to 2 + [2-4]

0 to 4 -> 0 to 3 + [3-4]

[1111, 112, 121, 13, 211, 22, 31]



```java
if(n == 0) {
    ArrayList<String>ba = new ArrayList<>();
    ba.add("");
    return ba;
}

if(n < 0) {
    ArrayList<String>ba = new ArrayList<>();
    return ba;
}

ArrayList<String>pnm1 = getStairPaths(n-1); //n-1 to 0 paths
ArrayList<String>pnm2 = getStairPaths(n-2); //n-2 to 0 paths
ArrayList<String>pnm3 = getStairPaths(n-3); //n-3 to 0 paths

ArrayList<String>myans = new ArrayList<>();

//n to 0 -> 1 + n-1 to 0
for(int i=0; i < pnm1.size();i++) {
    myans.add(1 + pnm1.get(i));
}

//n to 0 -> 2 + n-2 to 0
for(int i=0; i < pnm2.size();i++) {
    myans.add(2 + pnm2.get(i));
}

//n to 0 -> 3 + n-3 to 0
for(int i=0; i < pnm3.size();i++) {
    myans.add(3 + pnm3.get(i));
}

return myans;
```

[1111, 112, 121, 13, 211, 22, 31]

o