Recursion - 2

String str = "abc"
Find all the subsequences of str?
Subsequence is a string that
↳ 2^n subsequences.

"abc"
{"", "c", "bc", "b", "", "c", "ac", "a", "", "c", "b", "c"}

Subsequences are continuous.
Subsequences are not continuous.

"ab" → substr
"ac" → subsequence.

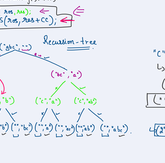
print → print("abc", "");

```
public static void print(String str, String res){
    if(str.length()==0){
        System.out.println(res);
        return;
    }
    char cc = str.charAt(0);
    String ros = str.substring(1);
    print(ros, res);
    print(ros, res+cc);
}
```

("abc", "")
↳ b, c

"c", ""
↳ c, ""
("" + c) res
("", "c")
""

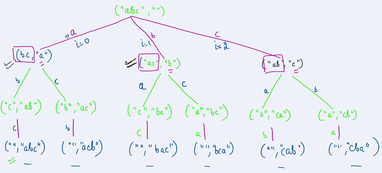
ros
"" + c → "c"



Recursion-tree

print permutations ("abc")

"abc" → "abc", "acb", "bac", "bca", "cab", "cba" → 6

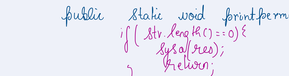


i=0 => "" + "bc" => "bc"
i=1 => "a" + "c" => "ac"
i=2 => "ab" + "" => "ab"

0
1
2

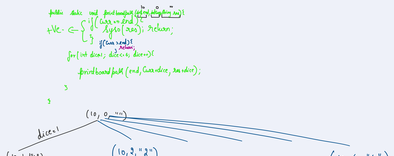
```
public static void printPermutations(String str, String res){
    if(str.length()==0){
        System.out.println(res);
        return;
    }
    for (int i=0; i<str.length(); i++){
        char cc = str.charAt(i);
        String ros = str.substring(0,i) + str.substring(i+1);
        printPermutations(ros, res+cc);
    }
}
```

Q. find land path



{(0,0), (0,1), (0,2), (1,2), (2,2)}
{(0,0), (1,0), (1,1), (1,2), (2,2)}
{(0,0), (1,0), (2,0), (2,1), (2,2)}
↳ print all ways to reach end from (0,0)?

```
public static void printWays(int sr, int sc, int dr, int dc){
    if(sr==dr && sc==dc){
        System.out.println(res);
        return;
    }
    if(canMove){
        for(int dir=0; dir<=2; dir++){
            printWays(nr, nc, dr, dc, res+dir);
        }
    }
}
```



tC => O(6^n)

Q.