

26 Oct

26 October 2022 21:35

Subsequence

abcd

↳ a
ab
ac
ad
abc
abd
acd

b
bc
bd
bcd

c d
cd

substring

abcd

ba

ab

abc

abcd

b
bc

bcd

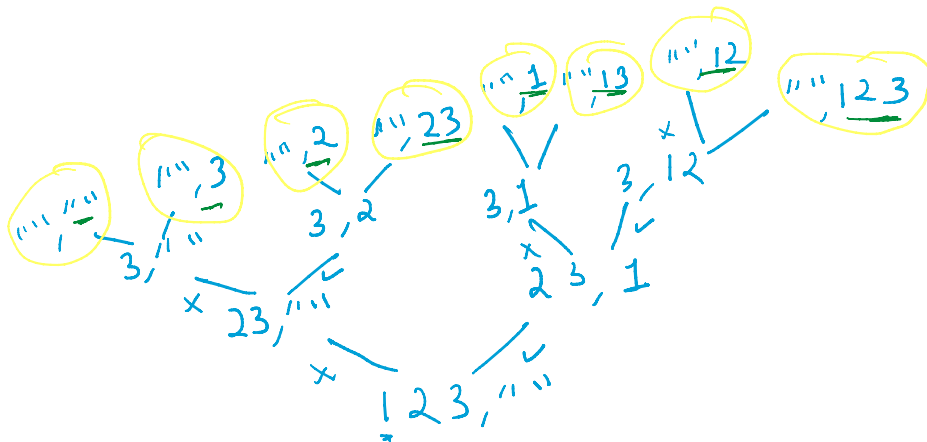
c d
cd

string $s = \underline{xyz}$

["", n, nx, nx², y, y², z]
xyz

[, 3, 2, 23, 1, 13, 12, 123]

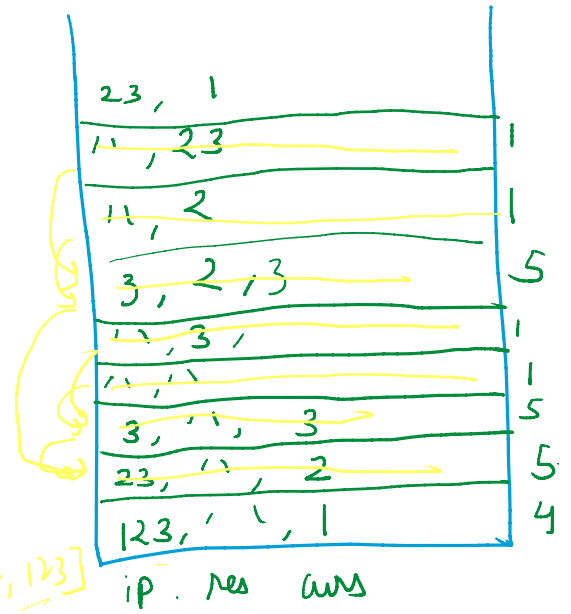
$\frac{2^n}{1}$



```

public static ArrayList<String> getALSubSeq(String ip, String res) {
1  if (ip.length() == 0) {
    ArrayList<String> currRes = new ArrayList<>();
    currRes.add(res);
    return currRes;
  }
2  char curr = ip.charAt(0);
  ArrayList<String> ans = new ArrayList<>();
3  ArrayList<String> rej =
    getALSubSeq(ip.substring(beginIndex: 1), res);
4  ArrayList<String> accept =
    getALSubSeq(ip.substring(beginIndex: 1), res + curr);
5  ans.addAll(rej);
  ans.addAll(accept);
  return ans;
}

```

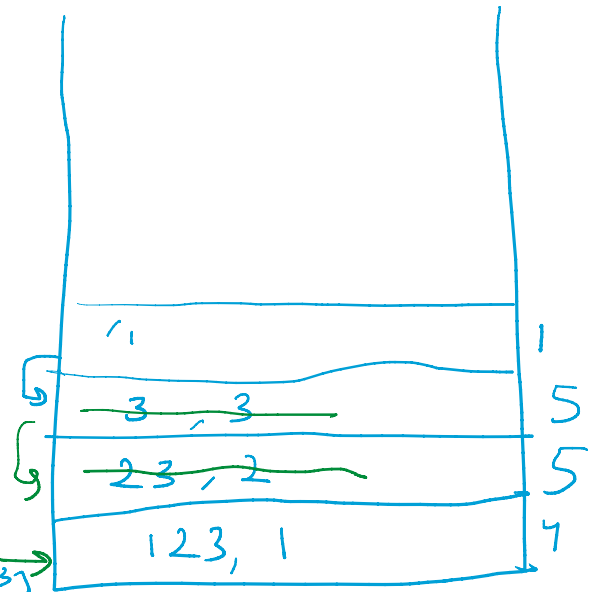


Handwritten notes: $[1, 3, 2, 23]$ and $[1, 13, 12, 123]$

```

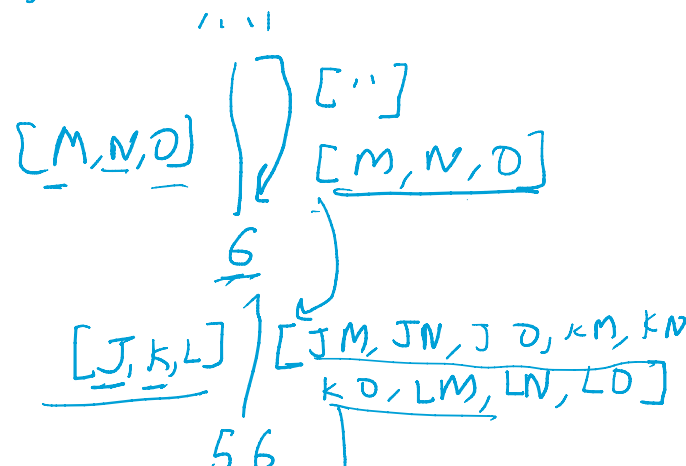
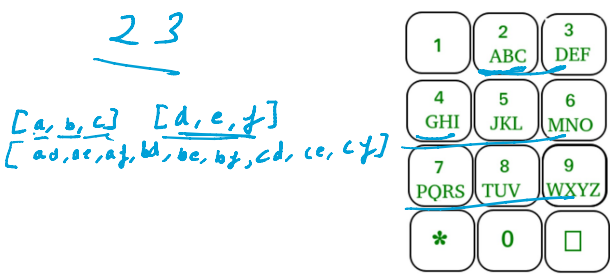
public static ArrayList<String> getALSubSeq2(String ip) {
1  if (ip.length() == 0) {
    ArrayList<String> currRes = new ArrayList<>();
    currRes.add("");
    return currRes;
  }
2  char curr = ip.charAt(0);
3  ArrayList<String> ans =
    getALSubSeq2(ip.substring(beginIndex: 1));
  int size = ans.size();
4  for (int i = 0; i < size; i++) ans.add(curr + ans.get(i));
5  return ans;
}

```



Handwritten notes: $[1, 3, 2, 3, 1, 13, 13, 123]$

Possible Words from Phone Digits:-



[

