Lecture 9

Saturday, 1 February 2020 2:30 PM

Recursion - 2 .

Q= print Subsequences . → "abc" → n → 2^n Subsequences .

a
b
c
ab
ac →
bc
abc
""

ab → 4

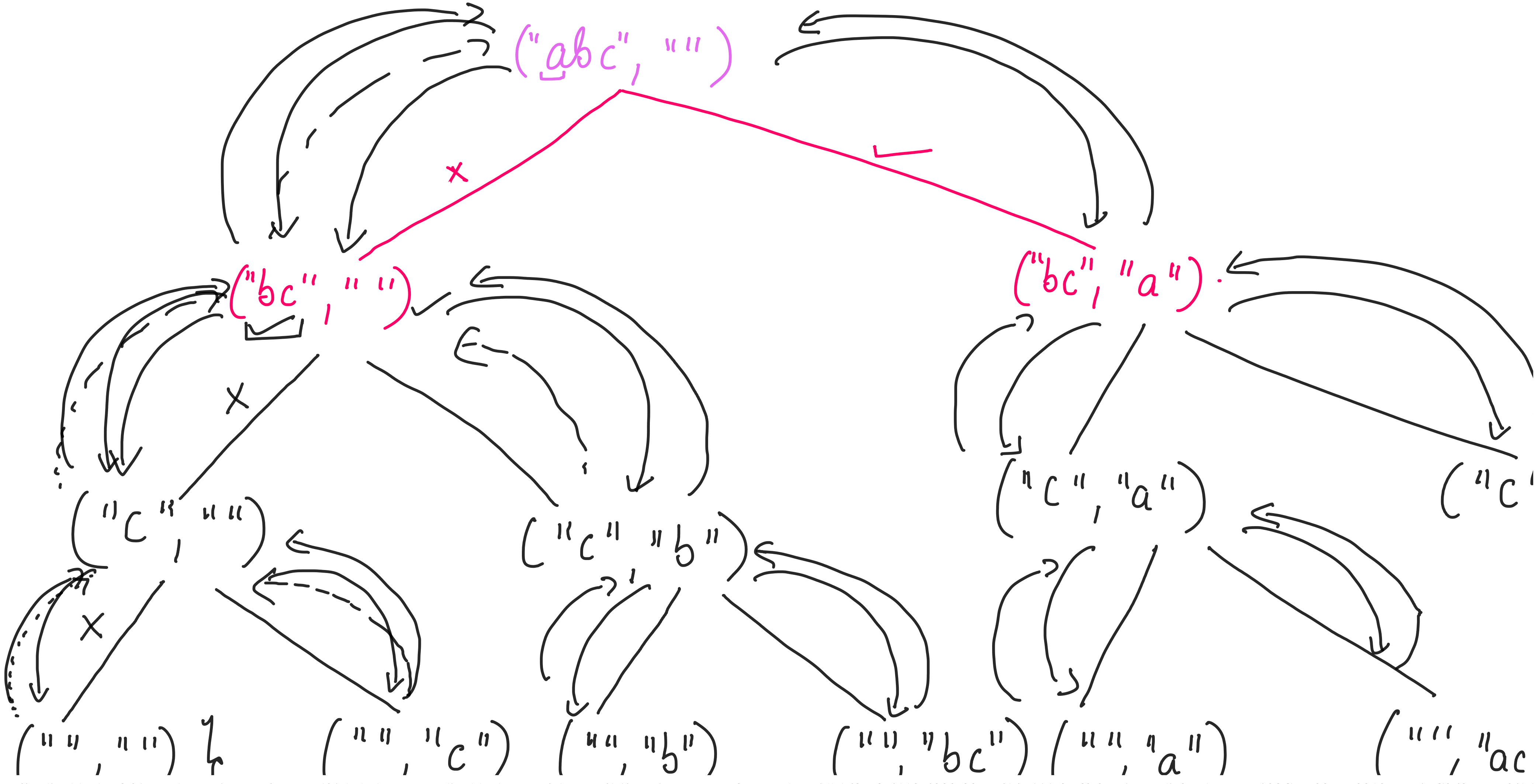
a
b
ab
""

```
                                        "abc"         ""
public static void printSubsequences(String str, String res){
        if (str.length() == 0){
                Syso(res);
                return;
        }
    char c = str.charAt(0);          //a.
    String ros = str.substring(1);  ✓  //bc ✓
    → printSubsequences(ros, res);
```

→ printsubsequences (ros, res);

└→ print ________ (ros, res+c);

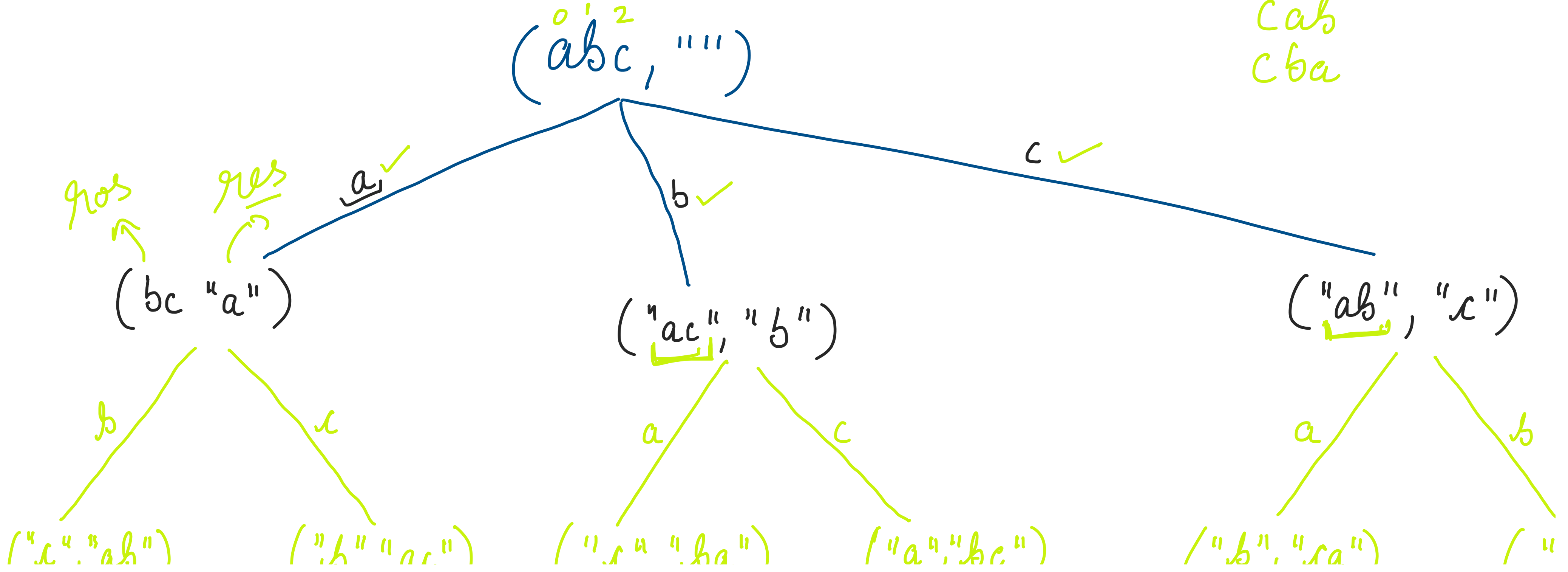
}

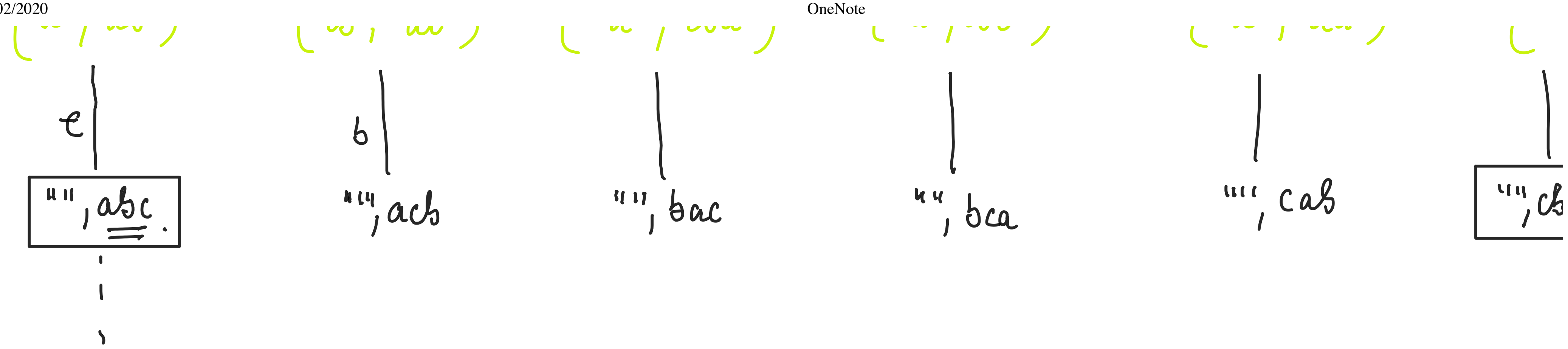


["", "c", "b", "bc"], "a", ac

✓ ✓ ✓ ✓ ✓ ✓

Q = print ~~permutations~~ → "abc" → { abc, acb, bac, bca, cab, cba





Q 3 => printBoardPath.

└> 0 → curr

10 → end

p s void pbp(int $\overset{10}{\overline{\overline{\text{end}}}}$, int $\overset{0}{\overline{\text{curr}}}$){

}

"6 4"

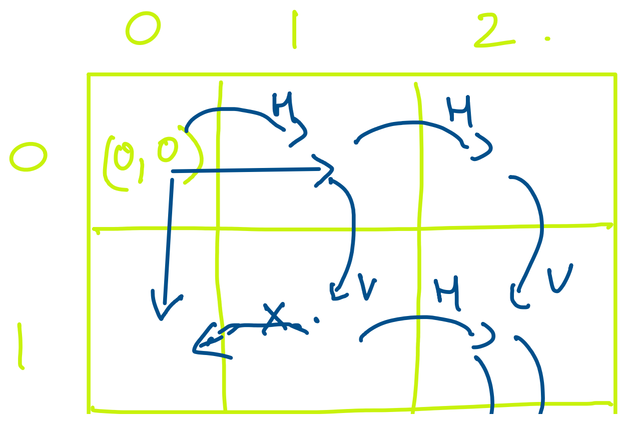
"5 3 2"

"2 5 3"

"2 3 5"

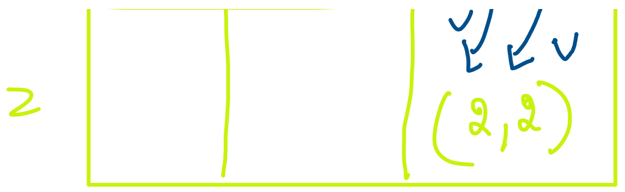
"3 3 3 1"

Q = print Maze Path.



p s pr

("HHV



"HVH
"HVV
"VHH
"VVH
"VHV

}

Q => printMazePathWD.

{ DD
DVH
VHHV
⋮
}

abc.

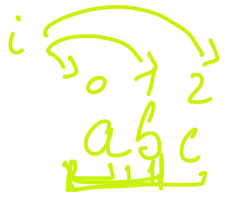
public static void printPermutations(String str,
String re

if (str.length()==0) {
Sysol(res);

```
        return;
    }

    for(int i=0 ; i< str.length(); i++){
        char cc= str.charAt(i);
        String ros = str.substring(0, i) + str.___(i+
        printper___(ros, res+cc);
    }

}
```



i=0 → "" + "bc" => "bc"

i=1 → "a" + "c" => "ac"

i=2 → "ab" + "" => "ab"

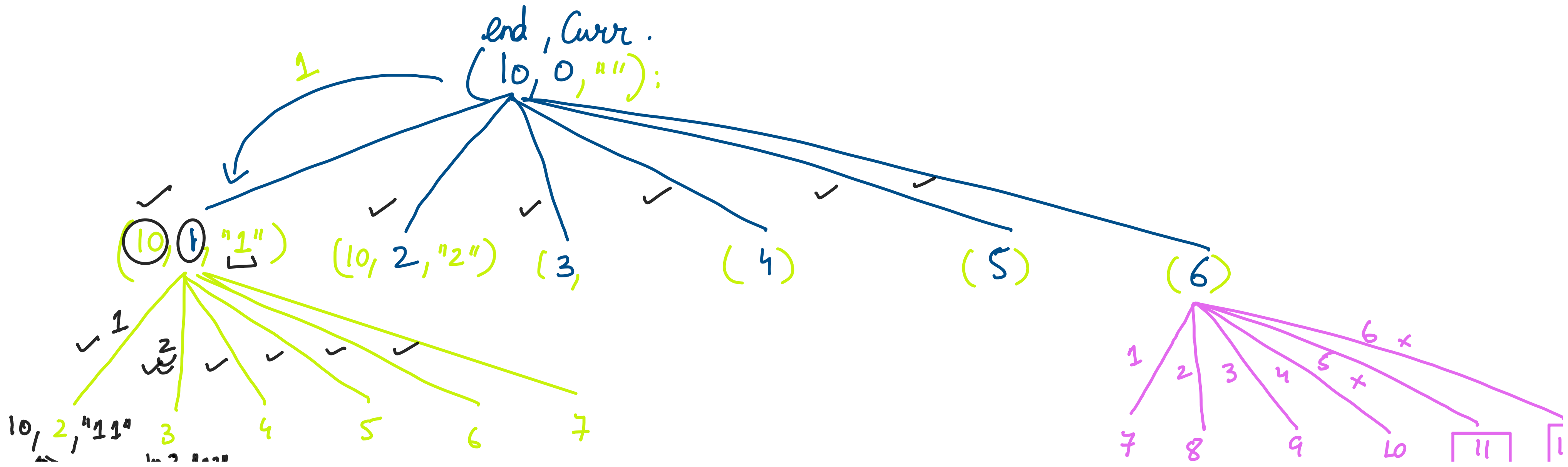
```
printboardpath(int end, int curr, String res){
    if (curr == end){
```

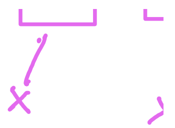
```
if (curr == end) {
    Syso(res);
    return;
}
if (curr > end) {
    return;
}
for ( int dice=1; dice<=6; dice++){

    pbp(end, curr+dice, res+dice);

}

}
```

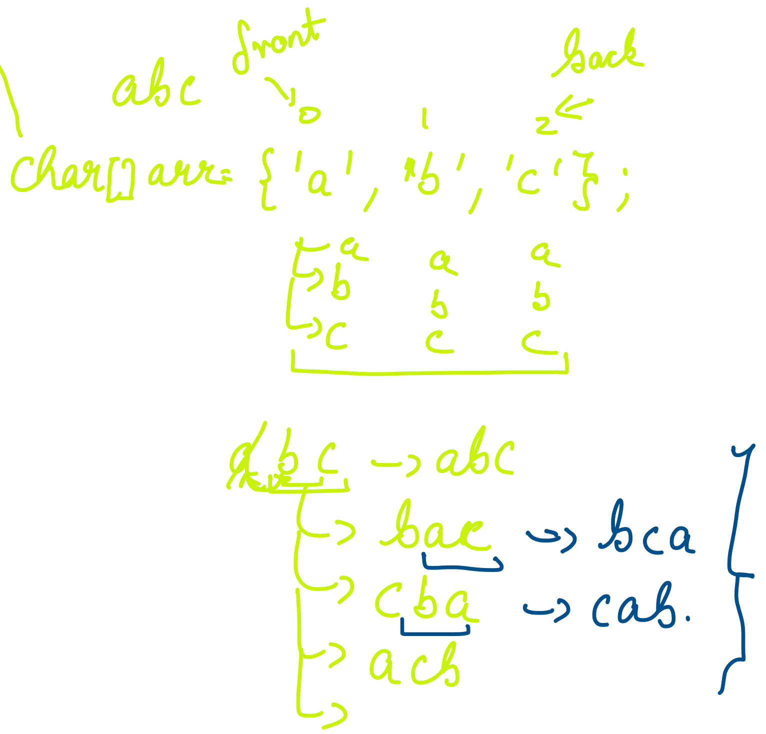




Q: ppws(char[] arr, int front, int back){

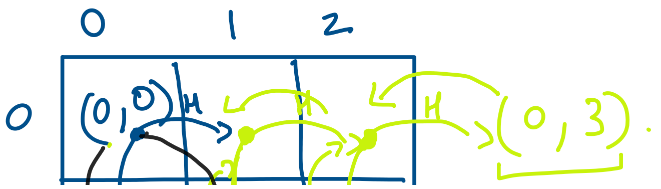
}

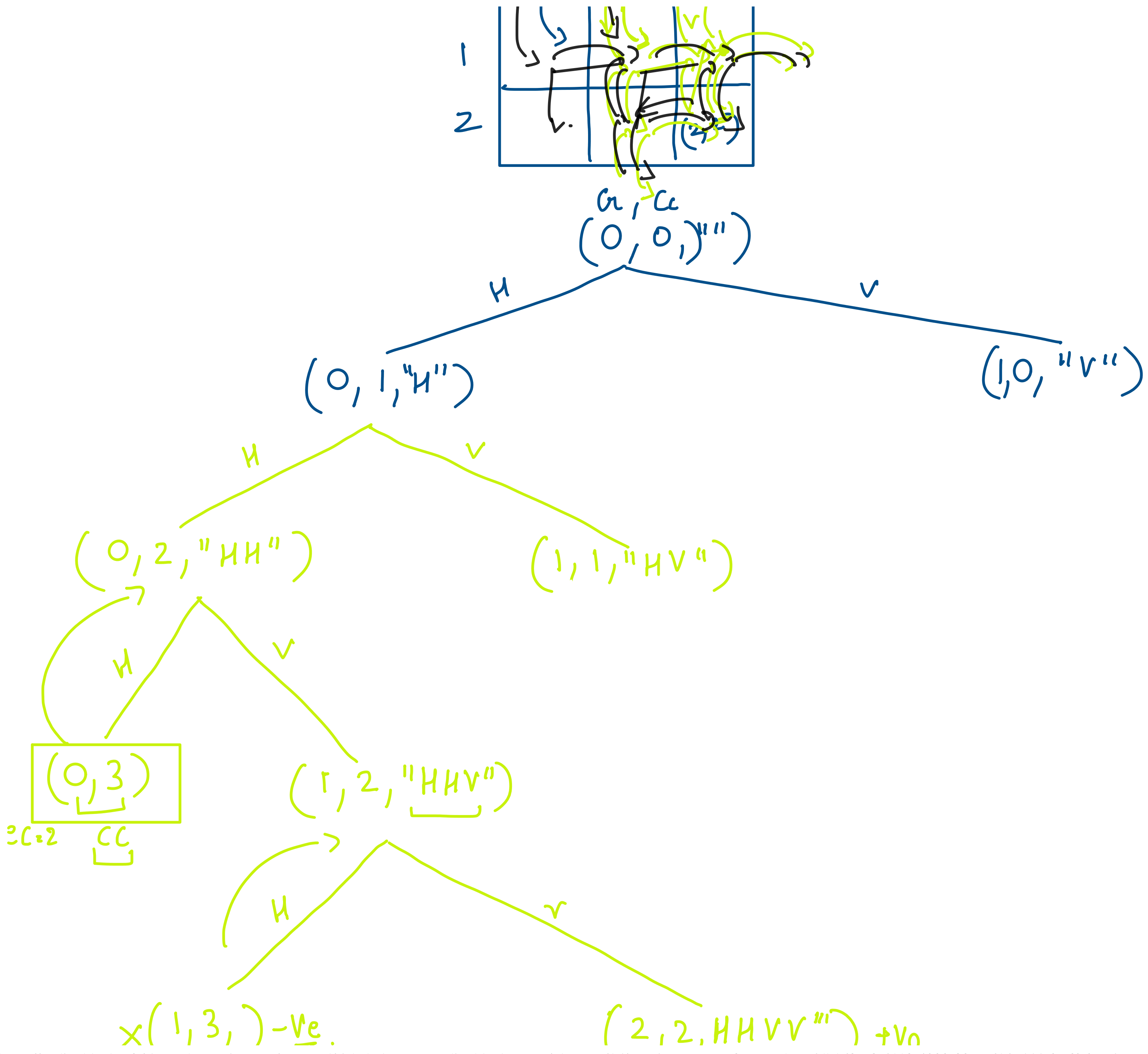
abc



```
p    s    void  printMP( int er, int ec, int cr, int cc, String res){
                       2       2       0       0                ""
        if (er==cr && ec==cc){
            syso(res)
            return;
        }
        if (cr>er || cc>ec){
            return;
        }
      printMP( er, ec, cr, cc+1, res+"H");
      printMP( er, ec, cr+1, cc, res+"V");

}
```

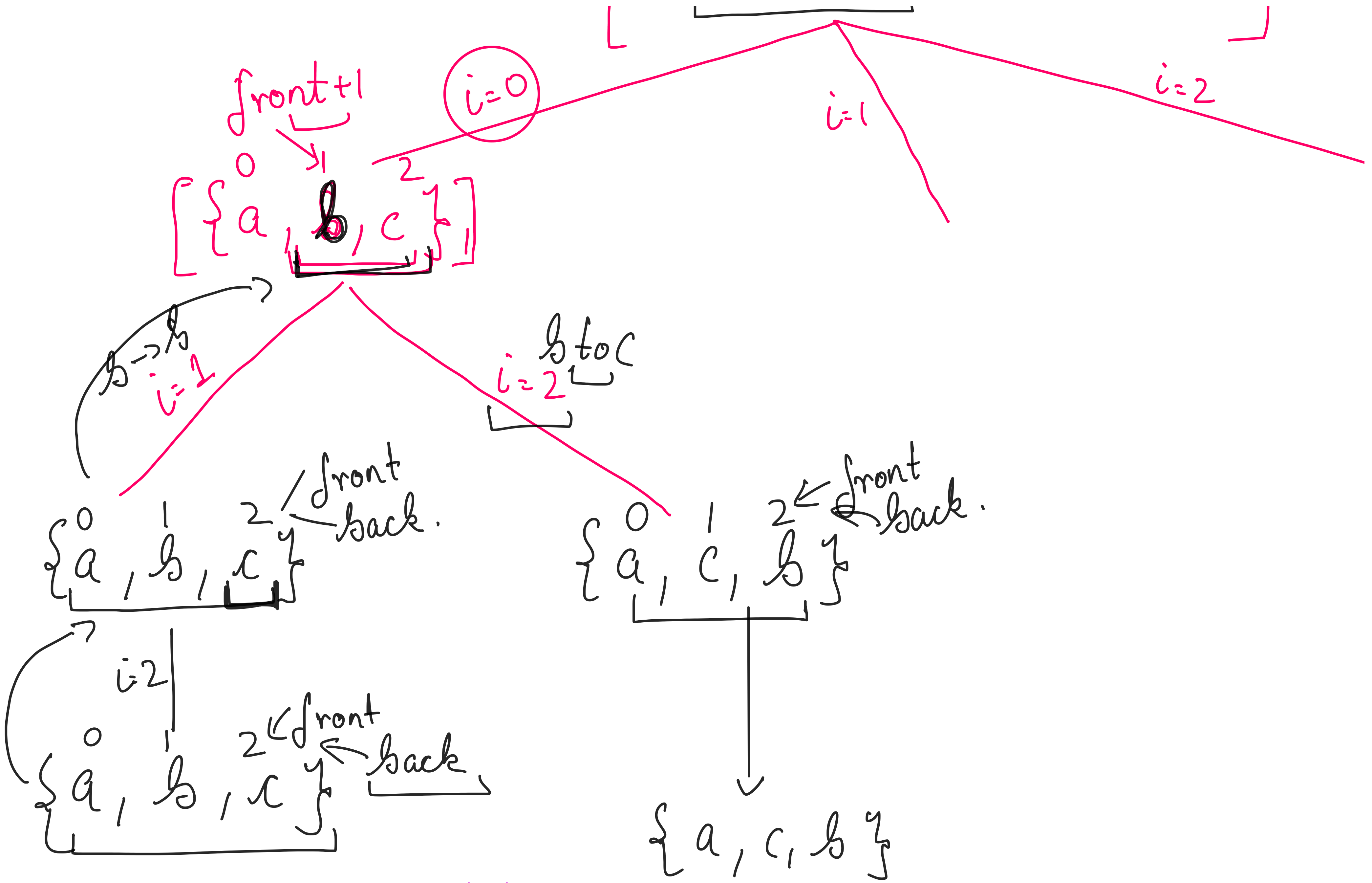



1
2
Gr , Cc
(0, 0, "")
H
V
(0, 1, "H")
(1, 0, "V")
H
V
(0, 2, "HH")
(1, 1, "HV")
H
V
(0, 3)
ec=2
cc
(1, 2, "HHV")
H
V
x(1, 3,) -ve.
(2, 2, HHVV"") +ve

```
                0   1   2
               {a , b , c }          0             2
ppws (char[] arr, int front, int back){
    if (front == back){
        display(arr);
    }

                   ~~0~~ 1      2
    for (int i=front; i<=back; i++){
                          1 , 2 {0, 1, 2}
        swap(arr, front, i);
                         2      2
        ppws(arr, front+1, back);
        swap(arr, front, i);  <=
    }
}

                  arr        front        back
    [ {a , b , c} ,    0   ,    2 ]
```



Facebook.

"1234"

```
public static void CodesOf String(String str, String res) {
    if (Str.length()==0){
        sysof(res)
        return;
    }
    String fc = str.substring(0,1);     // "1"
    String res1 = ______________(1);    // "234"
    char Code1 = getCode(fc);  <- a
    Codes of String(res1, res+Code1);  ✓
    if(str.length()>=2){
    String dc = str.substring(0,2);  "12"
    String res2 = ______________(2);  "34"
      if(Integer.value of(dc)<=26) {
                          ->12
          char Code2 = getCode(dc); <- l
          Codes of String(res2, res+Code2);
      }
    }
}
```

Codes

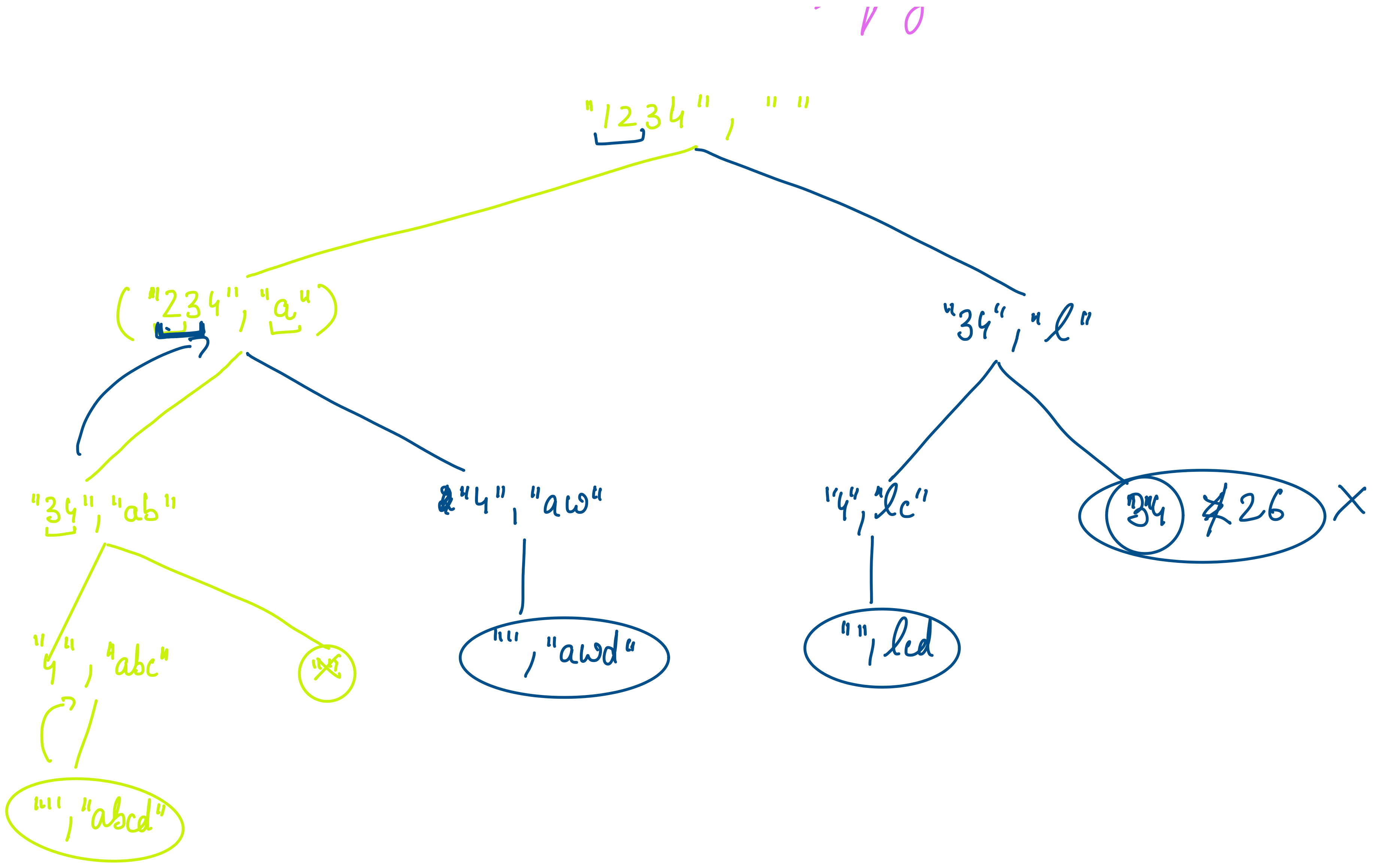
1 -> a
2 -> b
3 -> c
4 -> d
|
|
|
26 -> z.

"1234"

- -> abcd ✓ 1, 2, 3, 4
- -> lcd ✓ 12, 3, 4
- -> awd ✓ 1, 23, 4

1746

- -> a g d f
- -> q d f.



"1234" , " "
("234", "a")
"34", "l"
"34", "ab"
"4", "aw"
"4", "lc"
34 ~~4~~26
"4" , "abc"
"", "awd"
"", lcd
"", "abcd"