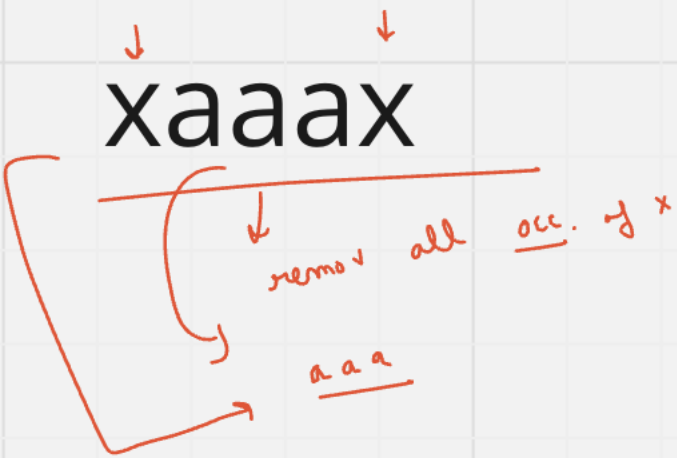
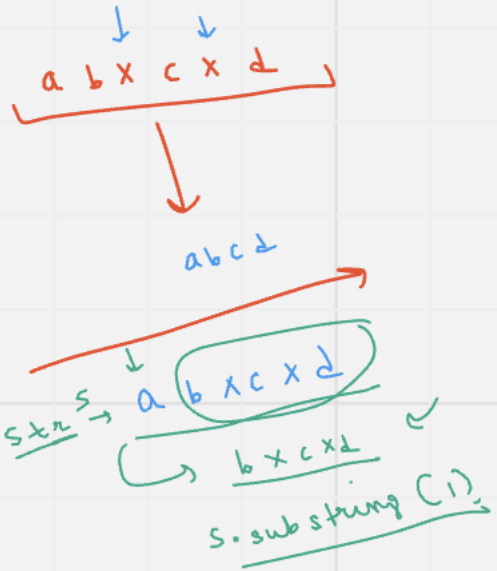
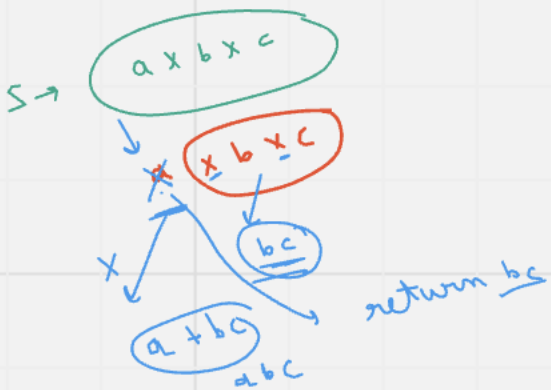
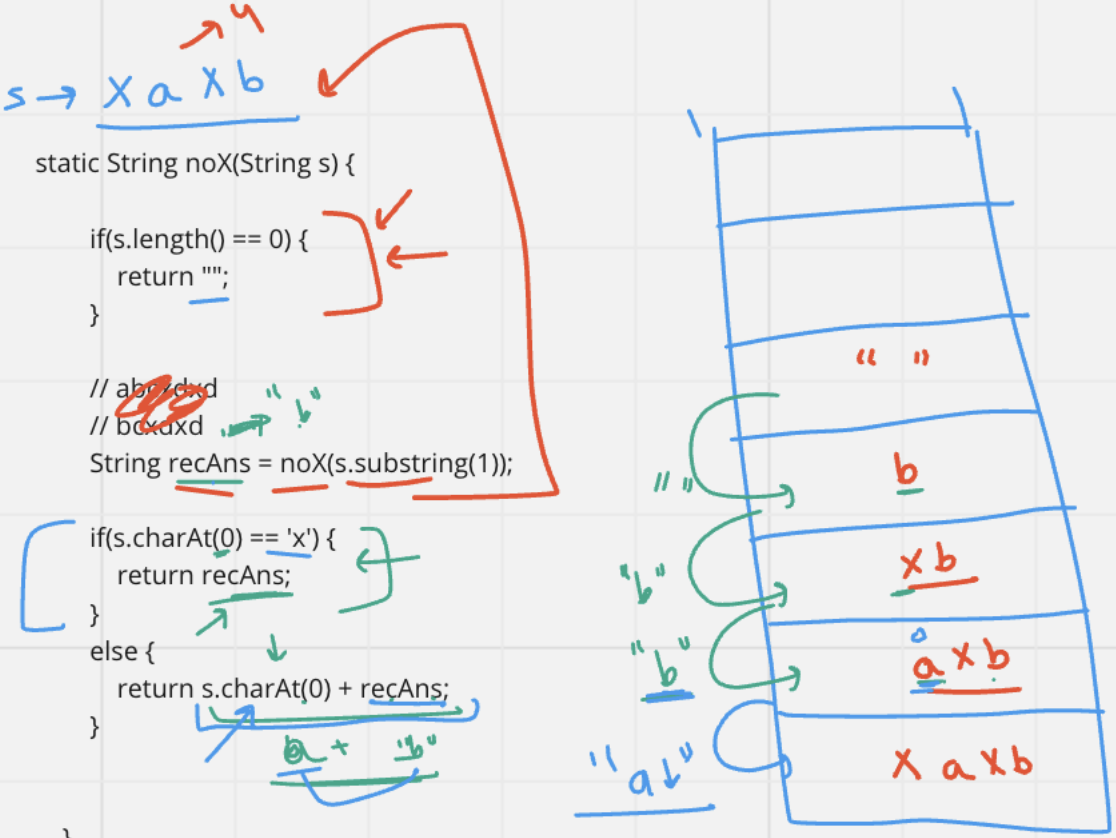
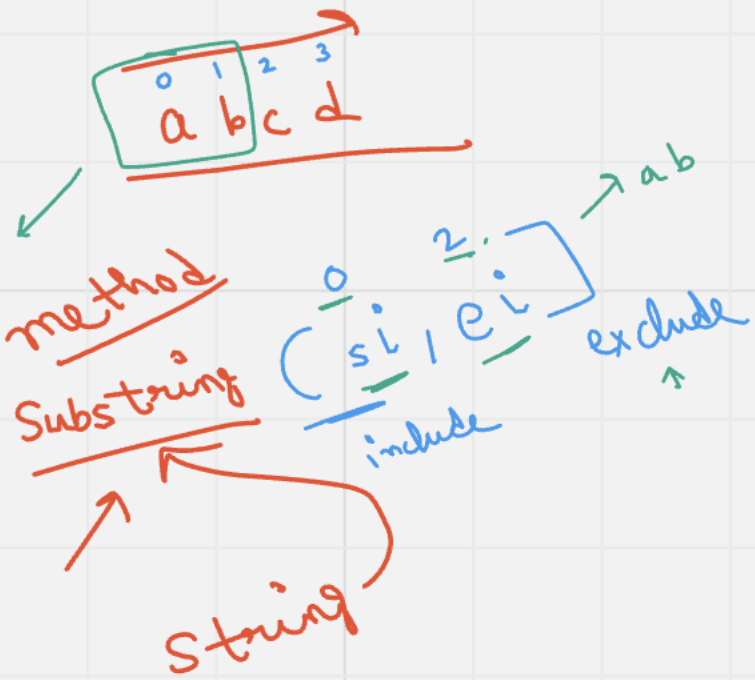
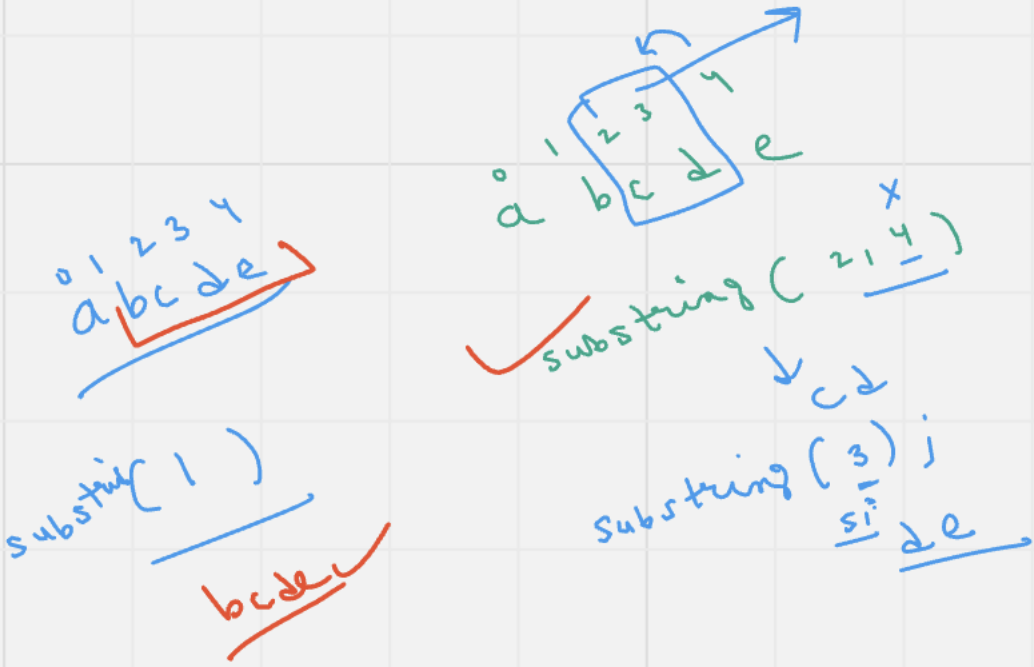

xaaax
remov all occ. of x
aaa

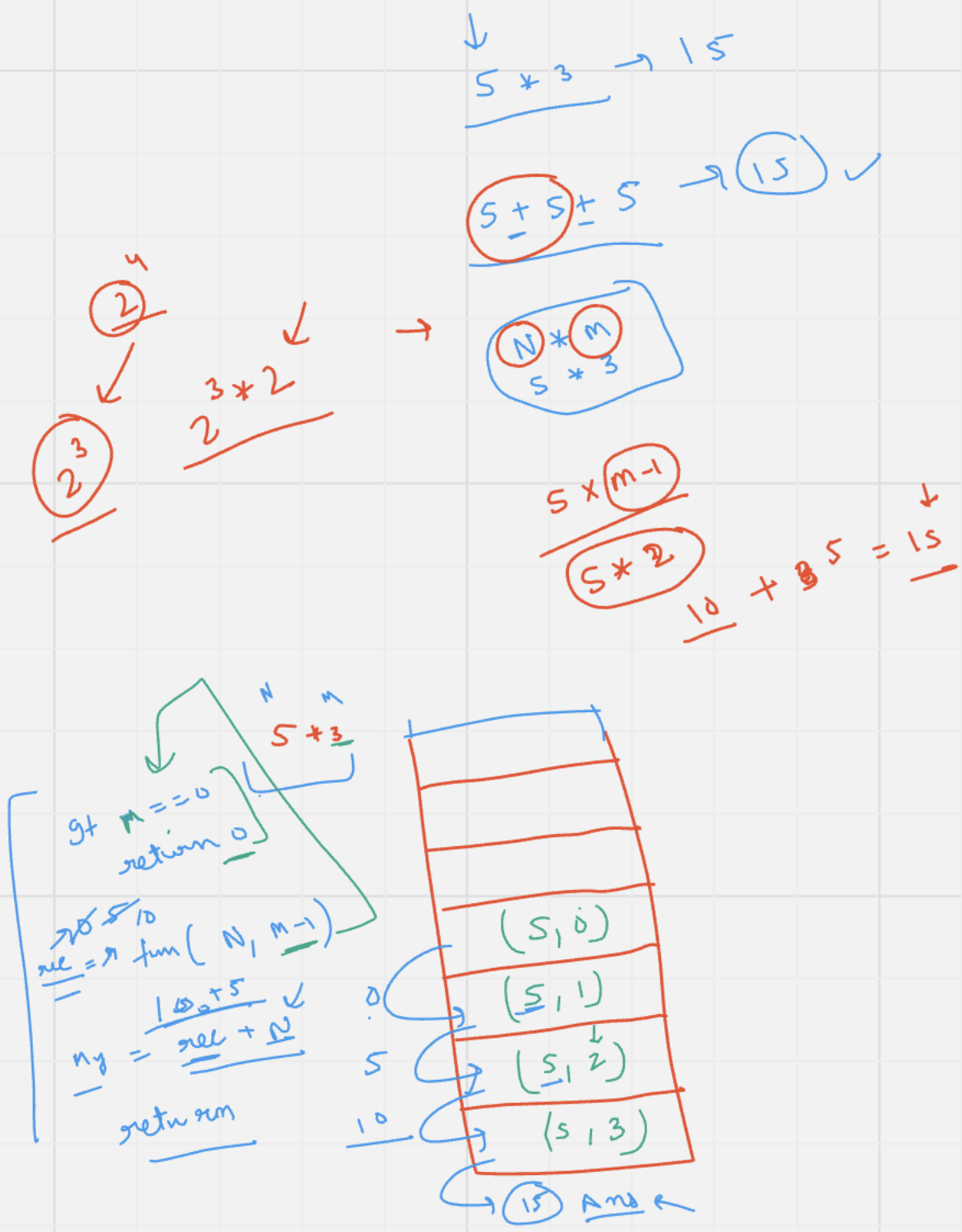

a b x c x d
abcd
str
a b x c x d
b x c x d
s.substring(1)

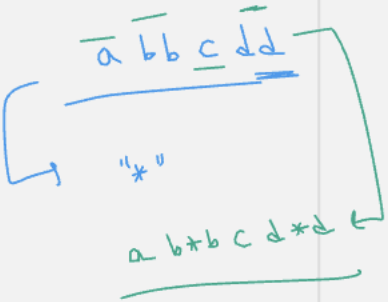

s
a x b x c
x b x c
x
bc
a + bc
abc
return bc


s → xaxb
static String noX(String s) {
if(s.length() == 0) {
return "";
}
// abxdxd
// bxdxd
String recAns = noX(s.substring(1));
if(s.charAt(0) == 'x') {
return recAns;
}
else {
return s.charAt(0) + recAns;
}
}
" "
b
xb
axb
xaxb
"b"
"b"
"a"

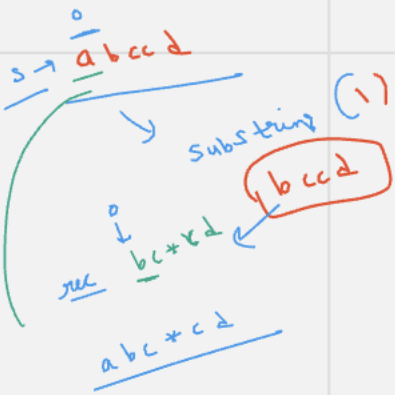

0 1 2 3
a b c d
method
Substring
(si , ei)
include
exclude
ab
String


0 1 2 3 4
a b c d e
substring (1)
bcde
0 1 2 3 4
a b c d e
substring (2 , 4)
cd
substring (3) ;
si
de


5 * 3 → 15
5 + 5 + 5 → 15
2^4
2^3
3 * 2
N * M
5 * 3
5 x m-1
5 * 2
10 + 5 = 15
N M
5 * 3
if m == 0
return 0
rec = fun(N, m-1)
my = rec + N
return
(5, 0)
(5, 1)
(5, 2)
(5, 3)
0
5
10
15 Ans

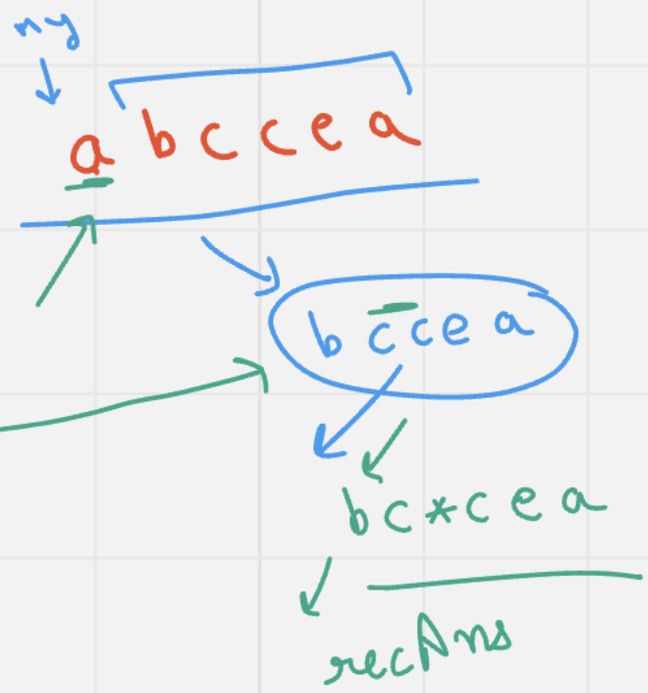

a bb c dd
"*"
a b*b c d*d


b b b
b * b * b


s
a b c c d
substring (1)
b c c d
rec
b c * c d
a b c * c d

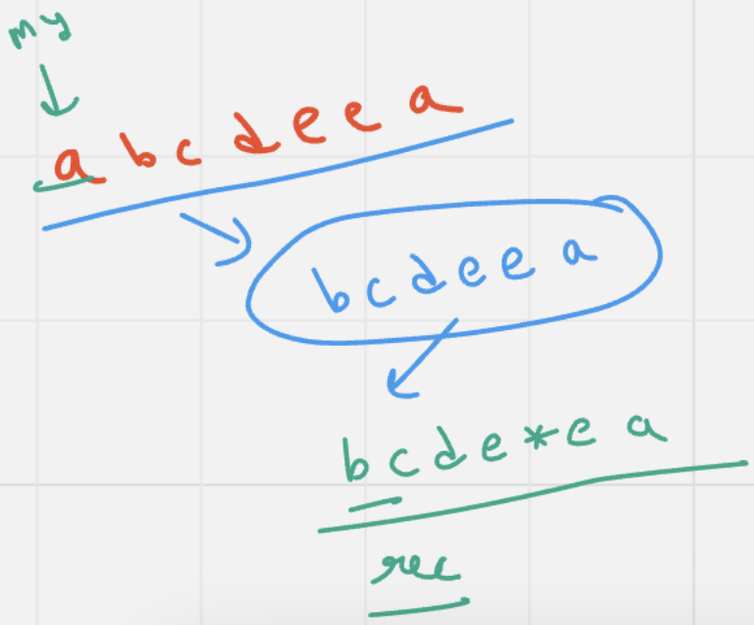

My
h h e e L o
h e e L o
rec
h e * e L o
h * h e * e L o

```
static String PairStar(String str) {
    //Write your code here

    if(str.length() == 1) {
       return str;
    }

    String recAns = PairStar(str.substring(1));

    if(str.charAt(0) == recAns.charAt(0)) {
       return str.charAt(0) + "*"+ recAns;
    }else {
       return str.charAt(0) + recAns;
    }

}
```

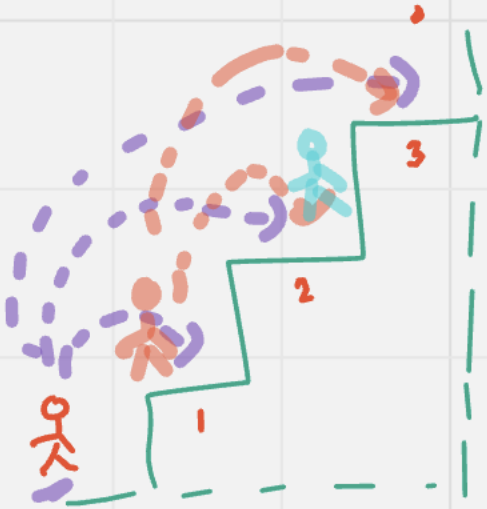


a + b c * c e a

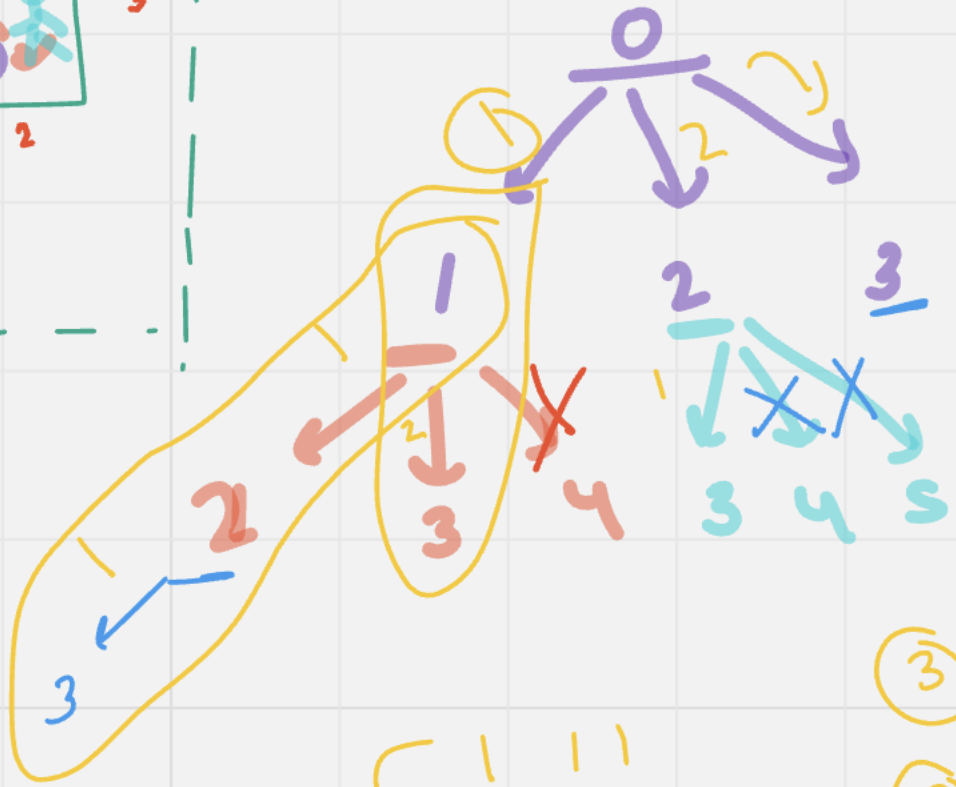
a b c * c e a ✓



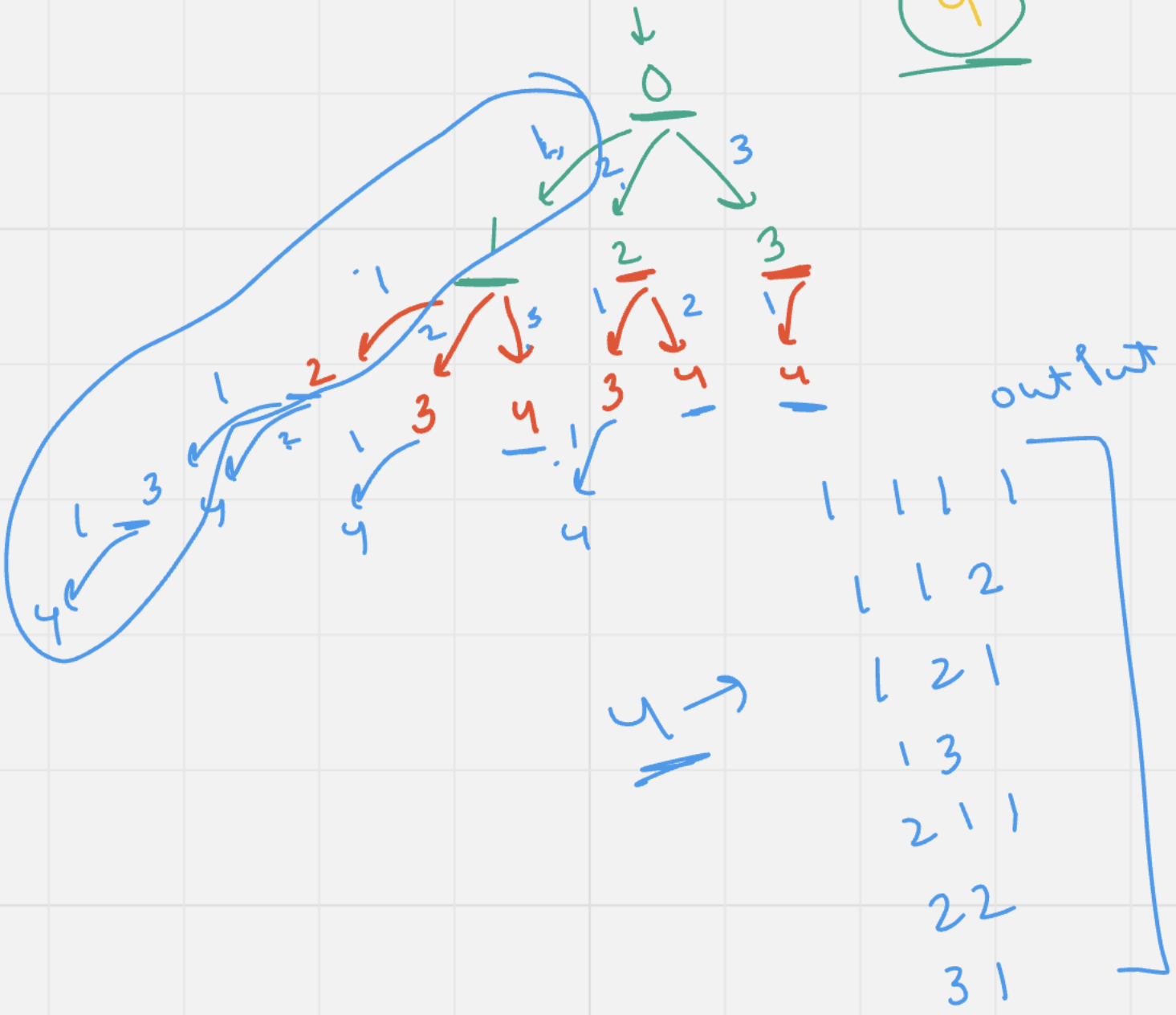
n → 3

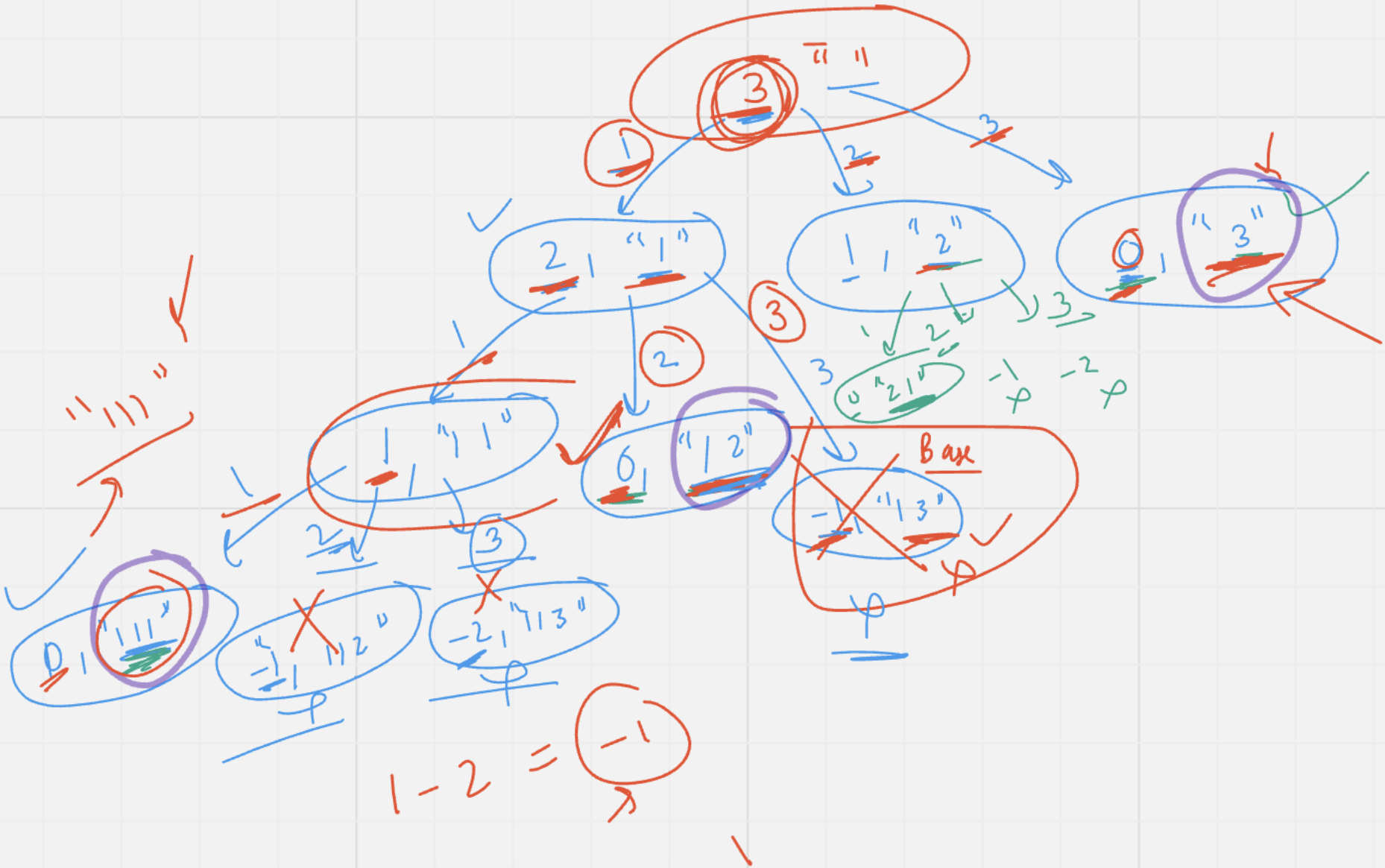


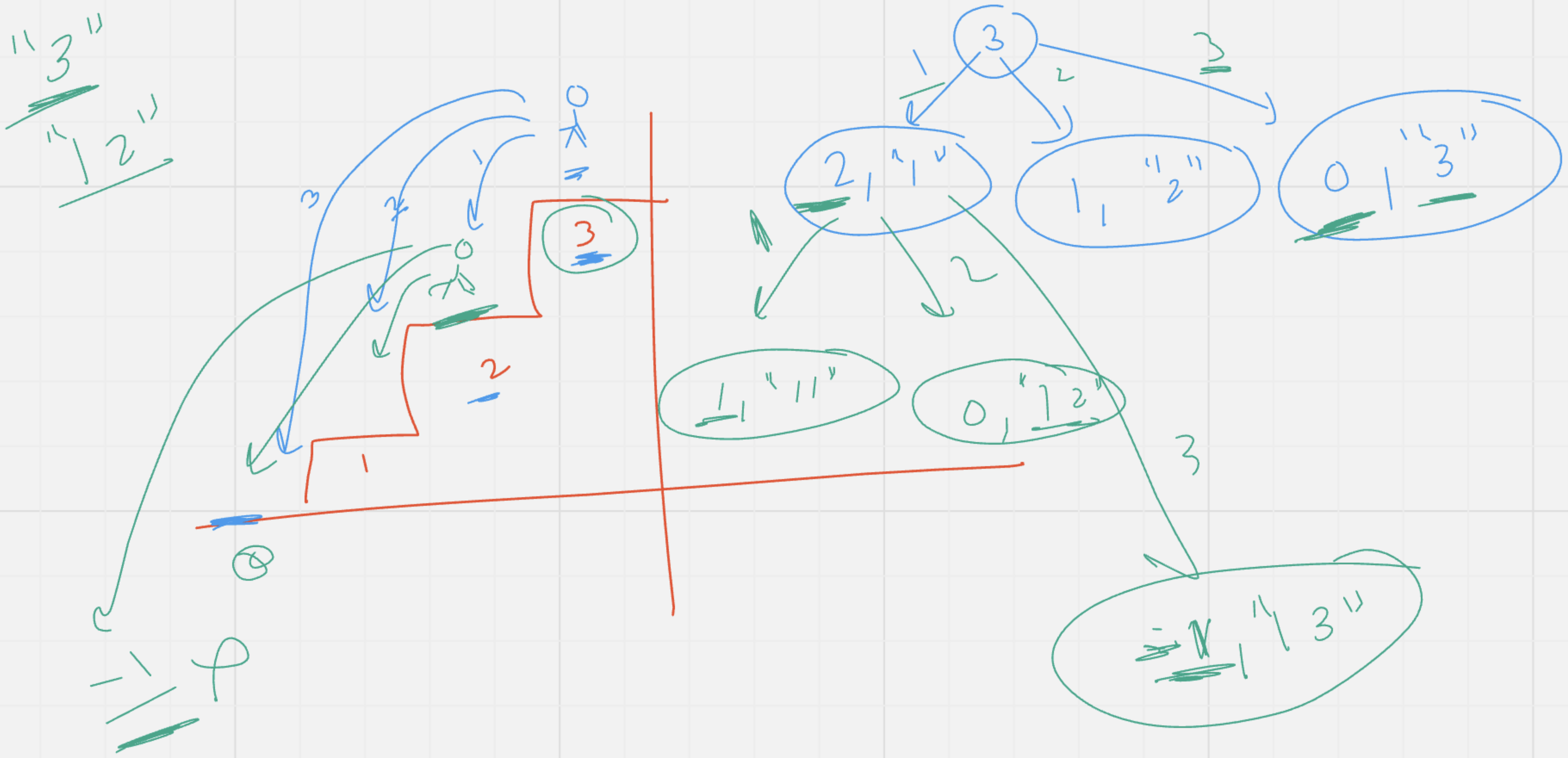
1 → 1 step
2 step
3 step



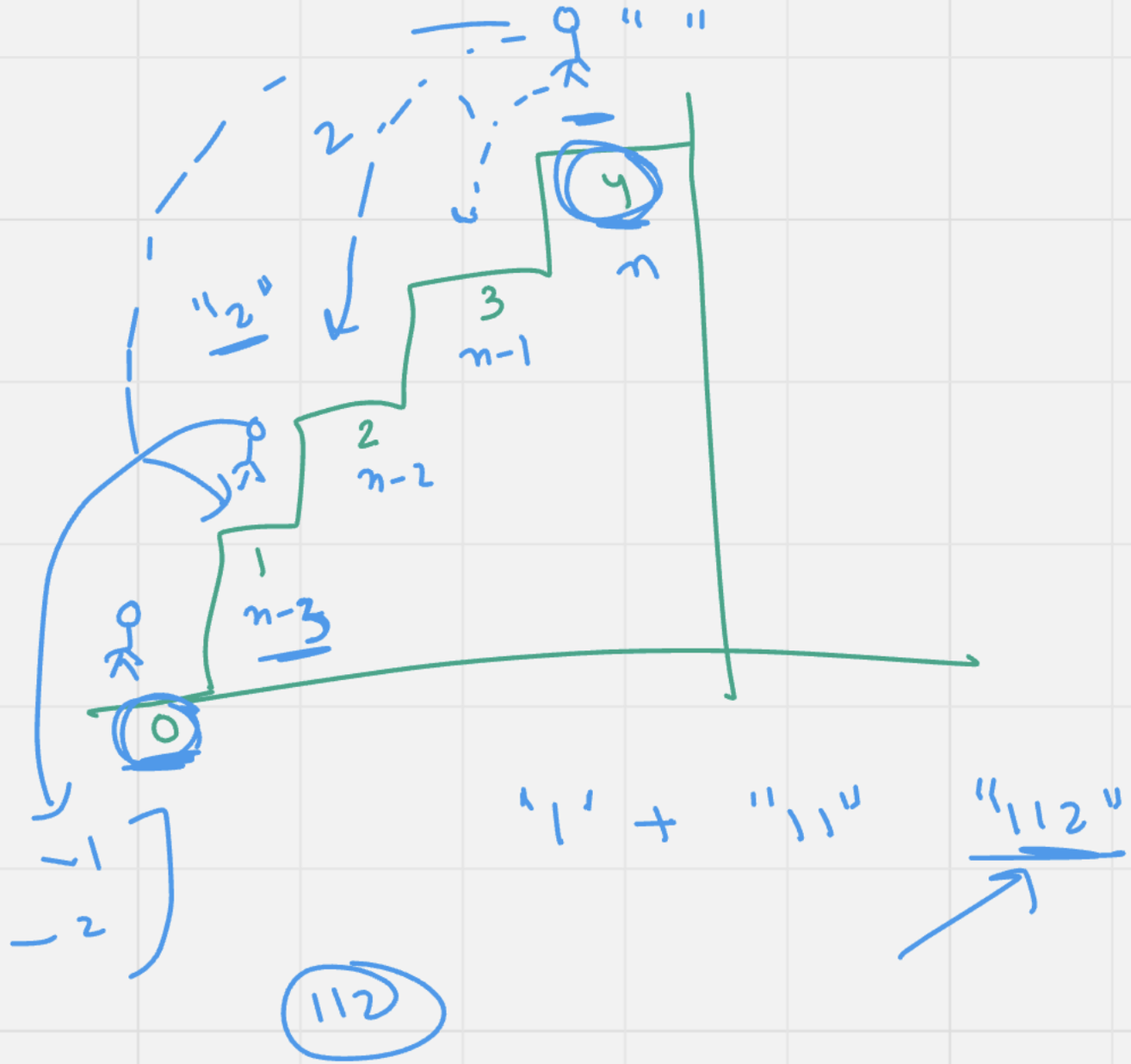
4 →

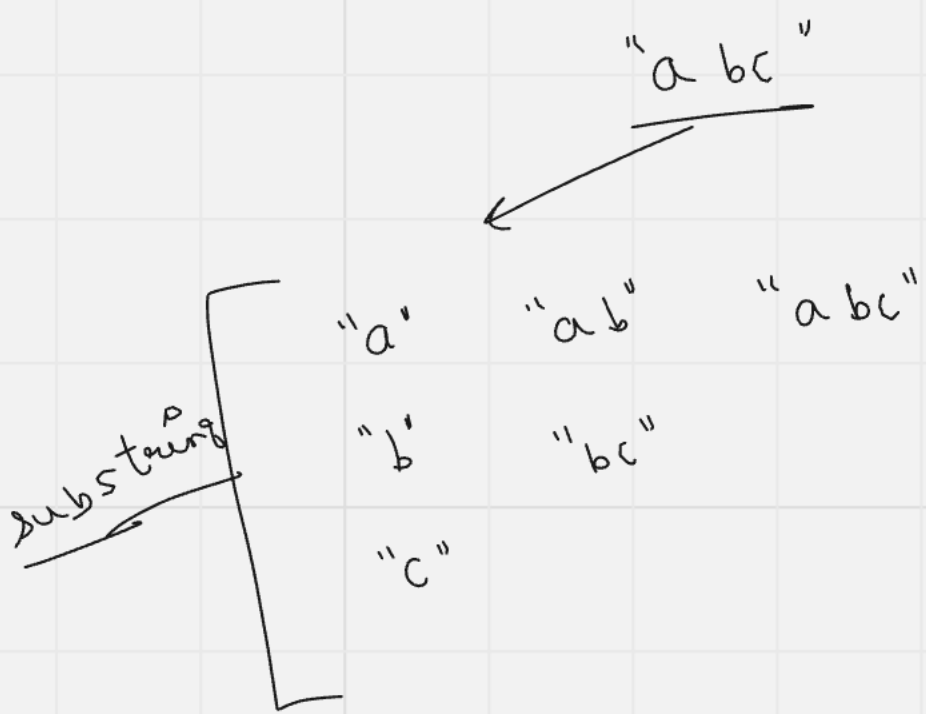

4
output
1 1 1 1
1 1 2
1 2 1
1 3
2 1 1
2 2
3 1

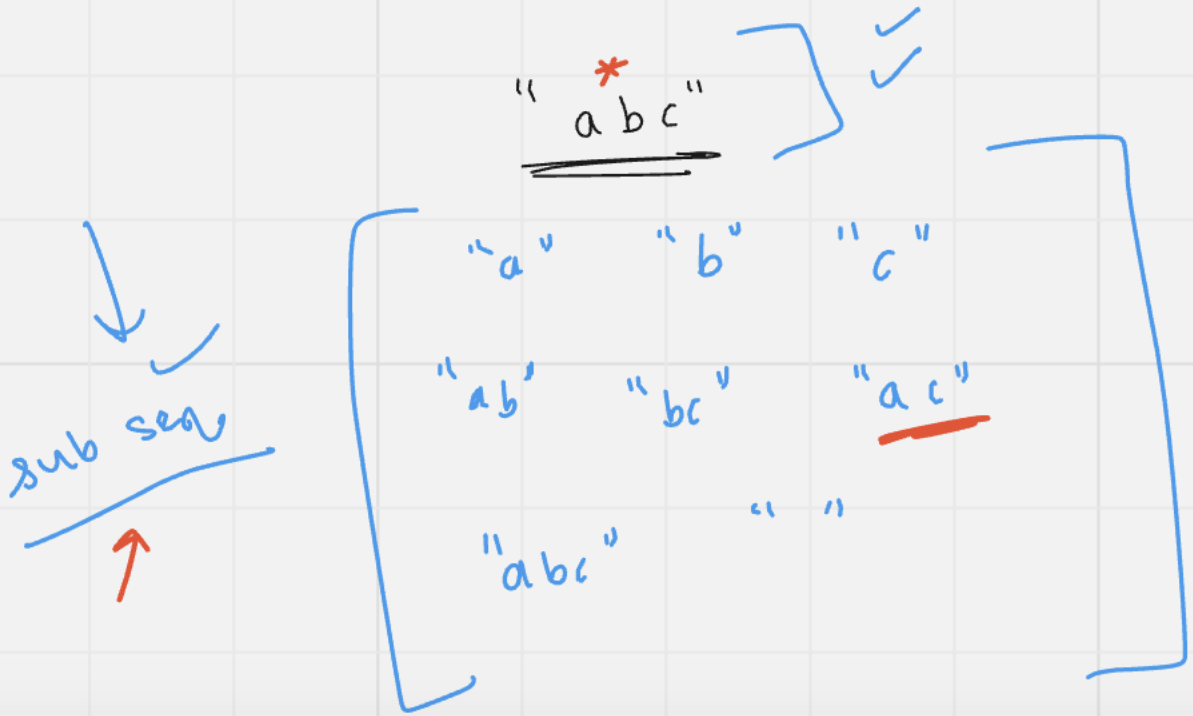

"111"
"11"
"1 1"
"12"
"13"
"21"
"2"
"3"
"1"
Вах
1 - 2 = -1

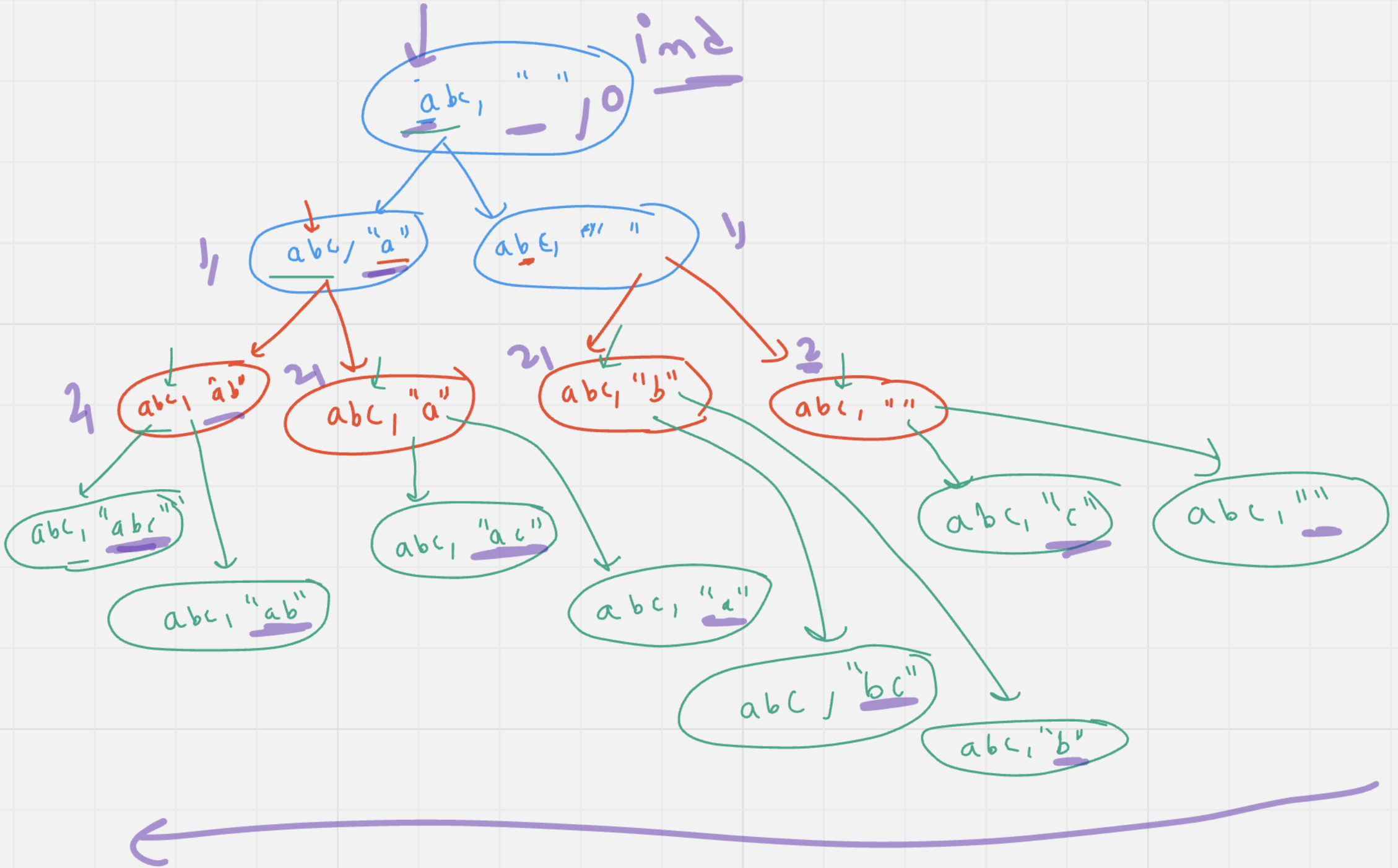

"3"
"12"
3
2
1
0
1
2
3
2, "1"
1, "2"
0, "3"
1, "11"
0, "12"
1, "3"

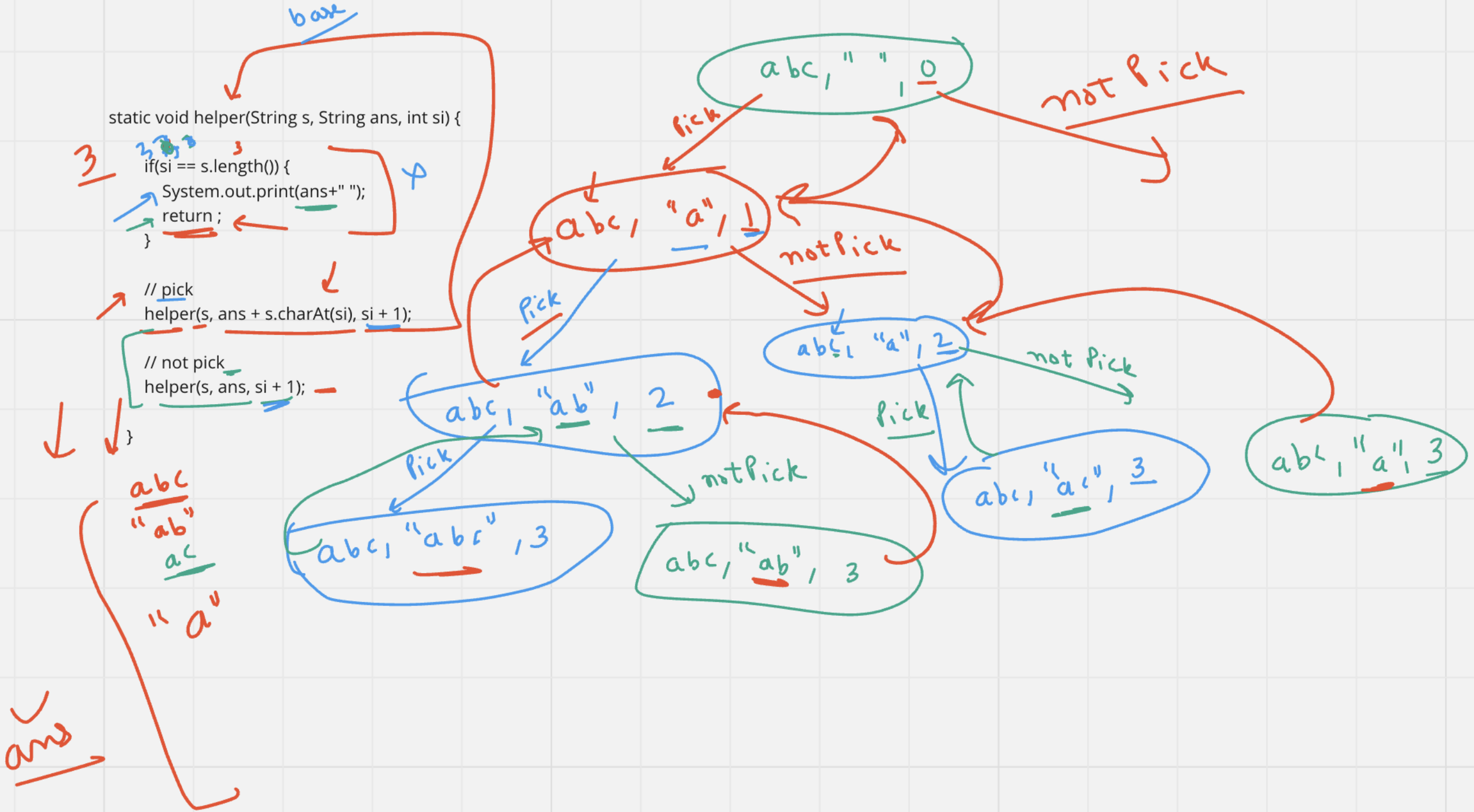
```
static void printStairPaths(int n,String pathSoFar) {

    if(n == 0) {
       System.out.println(pathSoFar);
       return ;
    }

    if(n < 0) {
       return ;
    }

    // one step
    printStairPaths(n-1, pathSoFar + "1");

    //two steps
    printStairPaths(n-2, pathSoFar + "2");

    //three steps
    printStairPaths(n-3, pathSoFar + "3");

}
```

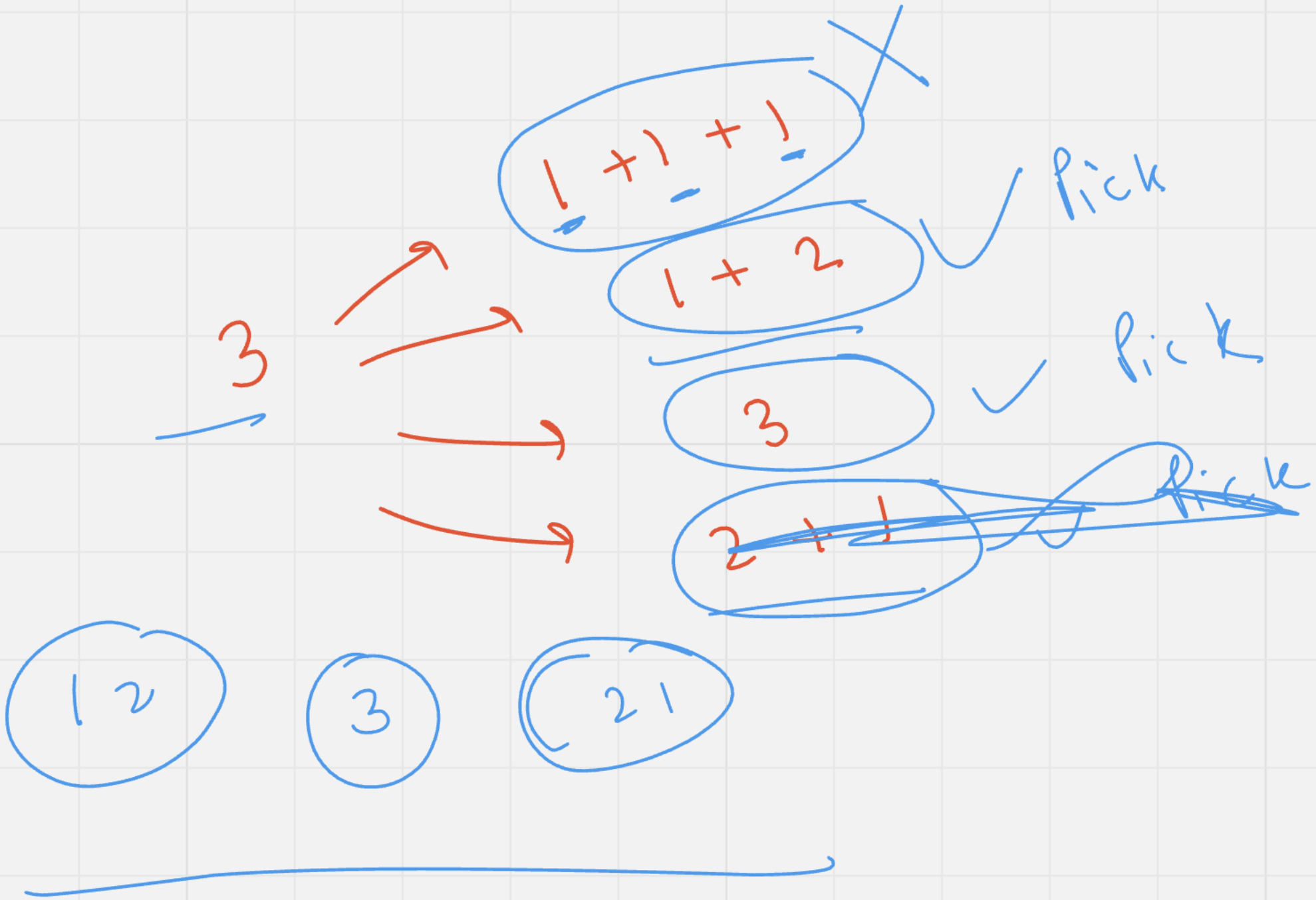


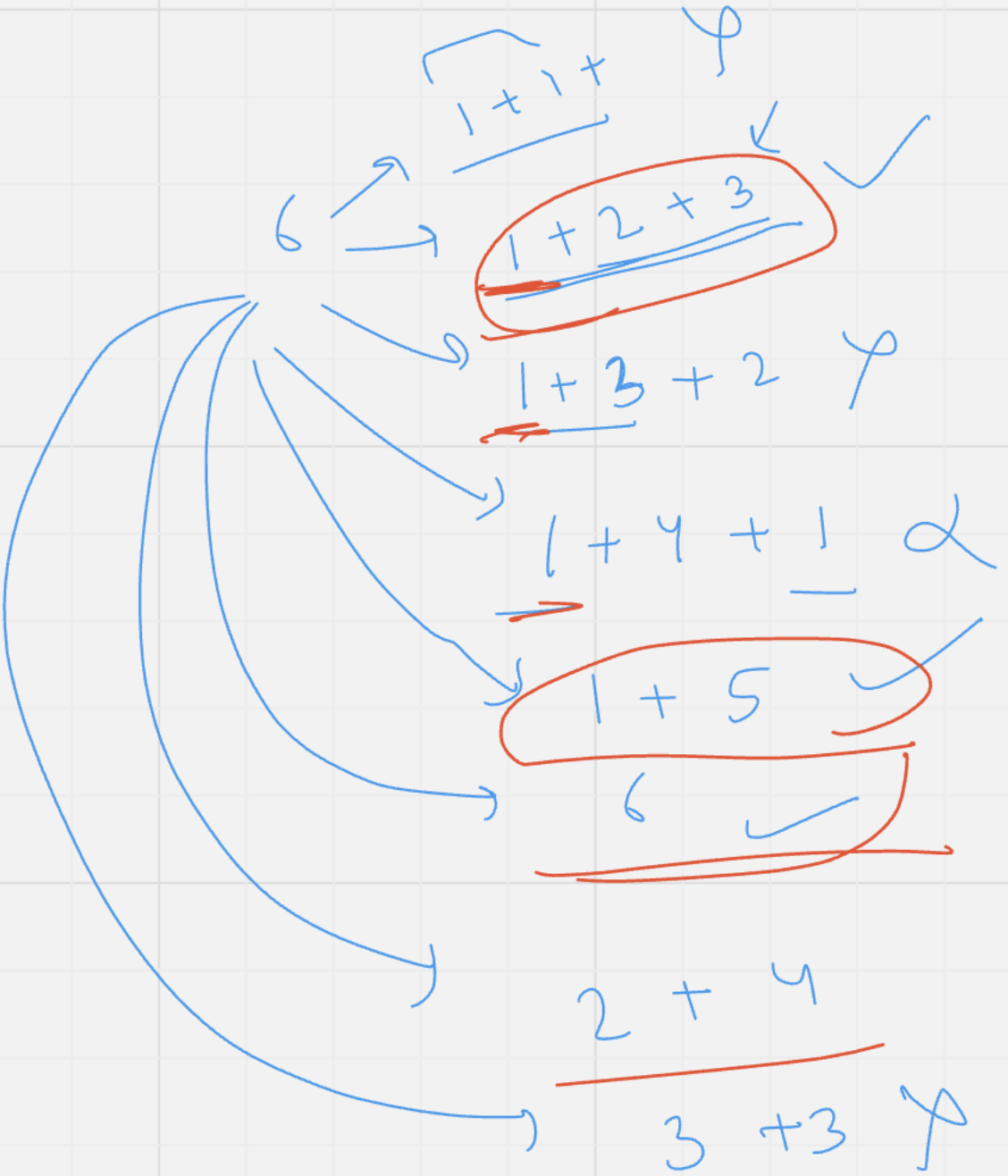

"abc"
substring
"a"
"ab"
"abc"
"b"
"bc"
"c"

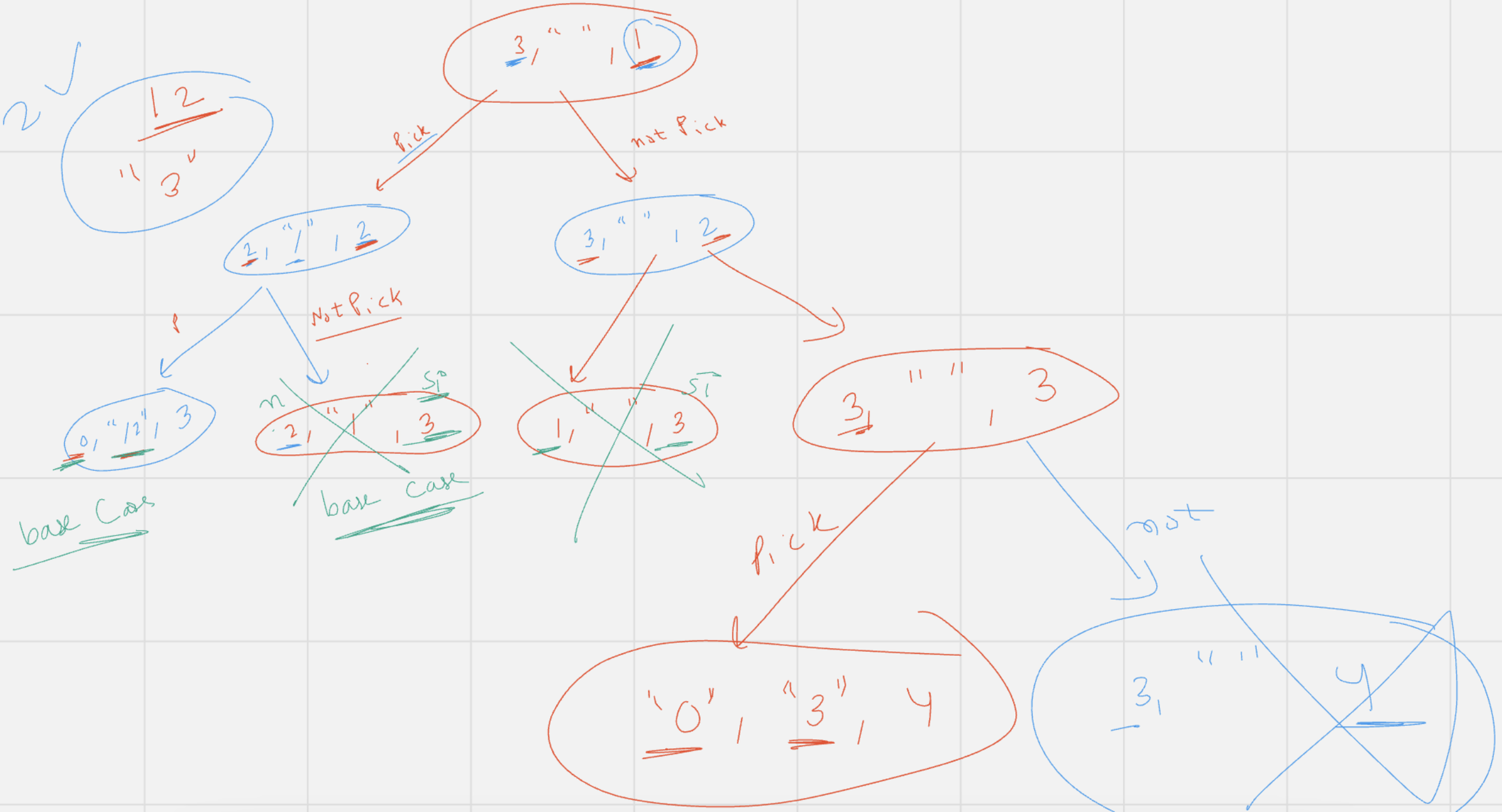

"abc"
subseq
"a"
"b"
"c"
"ab"
"bc"
"ac"
""
"abc"


ind
abc, "", 0
1
abc, "a"
abc, ""
1
2
abc, "ab"
abc, "a"
abc, "b"
abc, ""
2
2
2
abc, "abc"
abc, "ab"
abc, "ac"
abc, "a"
abc, "bc"
abc, "b"
abc, "c"
abc, ""


base
static void helper(String s, String ans, int si) {
if(si == s.length()) {
System.out.print(ans+" ");
return ;
}
// pick
helper(s, ans + s.charAt(si), si + 1);
// not pick
helper(s, ans, si + 1);
}
abc, " ", 0
pick
not Pick
abc, "a", 1
notPick
Pick
abc, "a", 2
not Pick
abc, "ab", 2
Pick
notPick
abc, "a", 3
abc, "ab", 3
abc, "abc", 3
abc
"ab"
ac
"a"
ans


3
1 + 1 + 1
1 + 2
3
2 + 1
Pick
Pick
Pick
1 2
3
2 1


1+1+
6
1+2+3
1+3+2
1+4+1
1+5
6
2+4
3+3


base Case
base case
pick
not pick
not