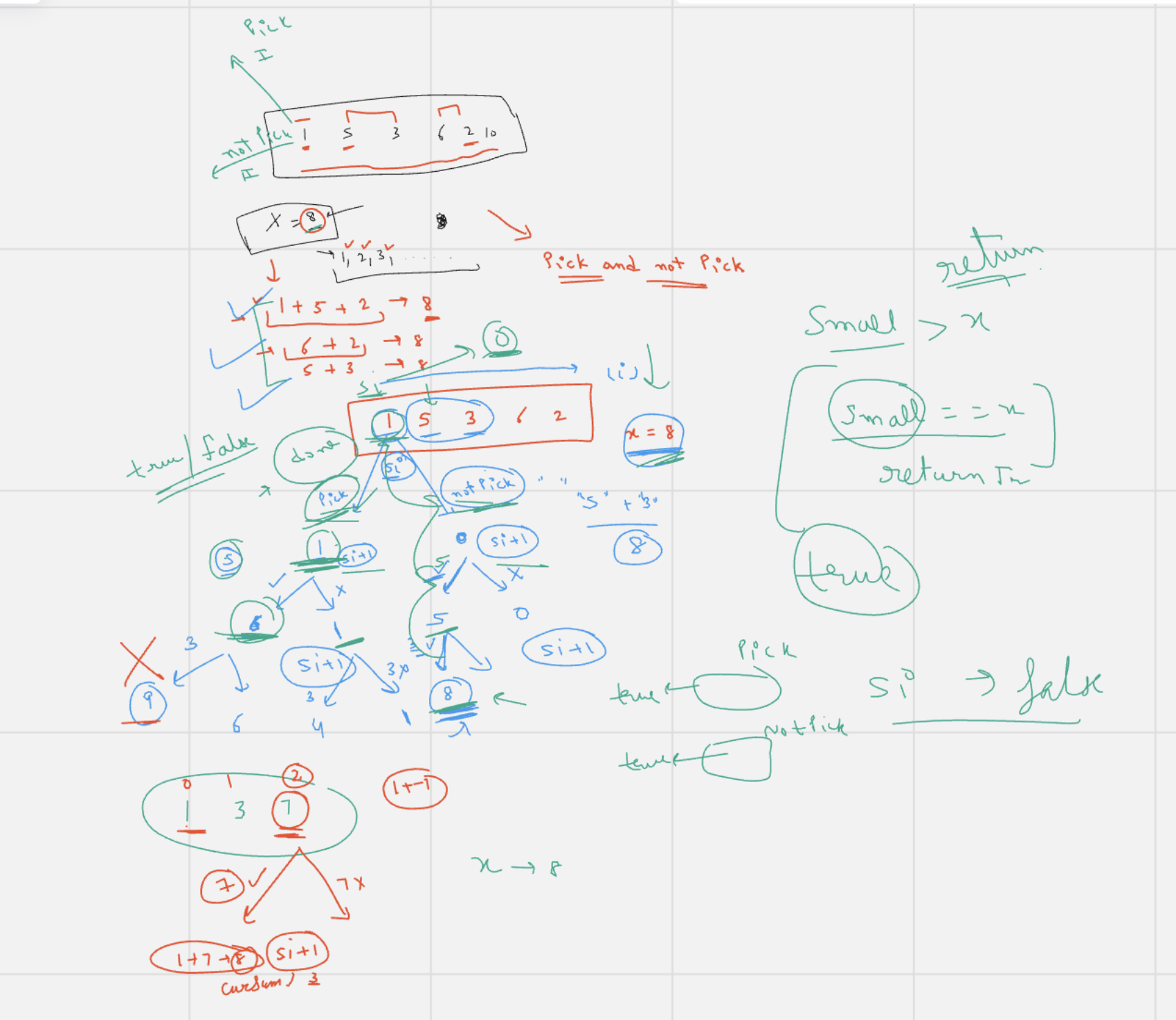
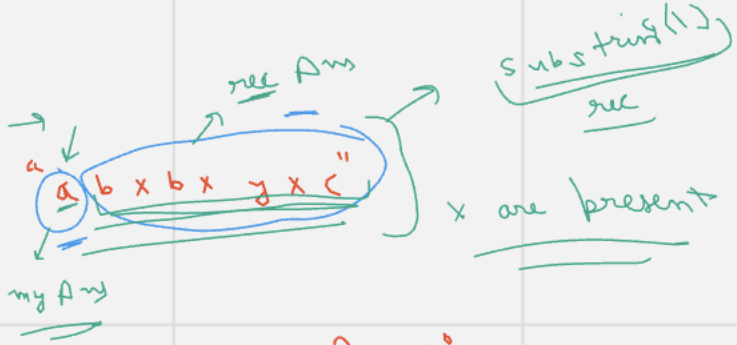

Pick
I
not Pick
II
1 5 3 6 2 10
X = 8
1, 2, 3,
Pick and not Pick
1 + 5 + 2 → 8
6 + 2 → 8
5 + 3 → 8
1 5 3 6 2
x = 8
true / false
done
Pick
not Pick
"5" + "3"
8
return
Small > x
Small == x
return True
true
Pick
true
not Pick
true
Si → false
x → 8
cursum = 3

Recursion [On Strings
is 2D Array]

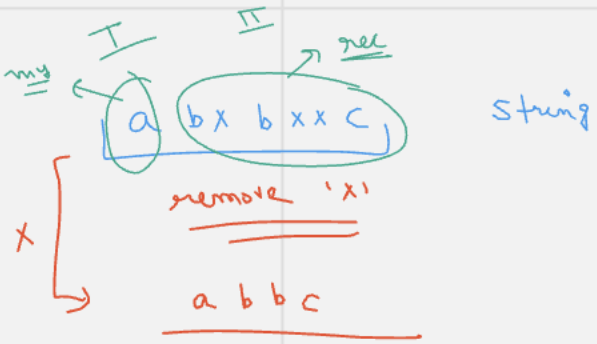


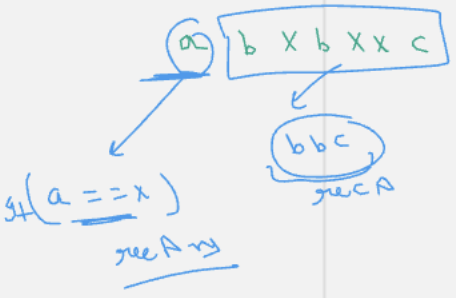
String function

① [
length()
charAt(ind)
substring(si, ei) — not include

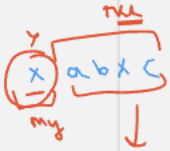
s = a b c d (0 1 2 3)

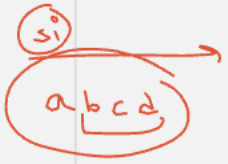
✓ [
s.charAt(2) → c
s.substring(1,3) → bc
s.substring(2) → cd
(1) → bcd





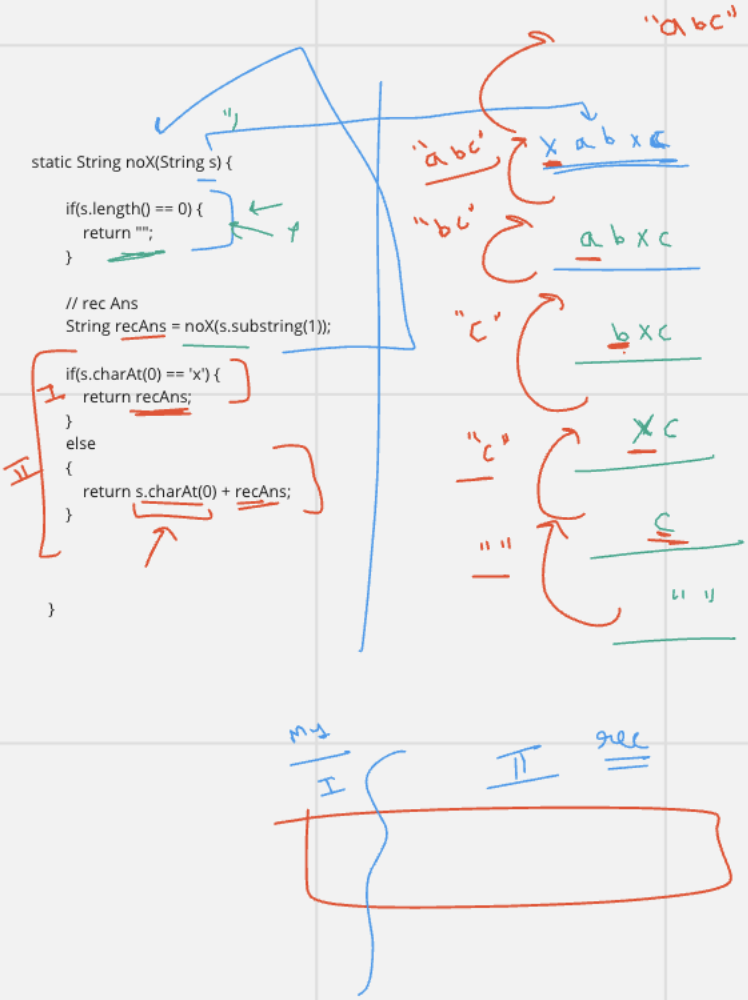

```
}   a + recAns
}   a + bbc = abbc
```





s. substring(1)

↳ bcd


static String noX(String s) {
if(s.length() == 0) {
return "";
}
// rec Ans
String recAns = noX(s.substring(1));
if(s.charAt(0) == 'x') {
return recAns;
}
else
{
return s.charAt(0) + recAns;
}
}
I
II
"abc"
"bc"
"c"
""
"abc"
xabxc
abxc
bxc
xc
c
""
my
I
II
rec

hello

hel*lo

xx y zzz wp

x*x yz*z*z wp

"h"

I II rec

hello

rec

hel*lo

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

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

if(str.charAt(0) == str.charAt(1)) {
    return str.charAt(0) + "*" + recAns;
}
else
{
    return  str.charAt(0) + recAns;
}
```

hello

rec → hel*lo

h + * + hel*lo

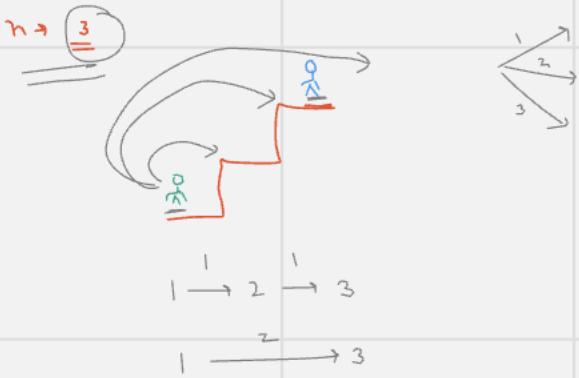
h*hel*lo

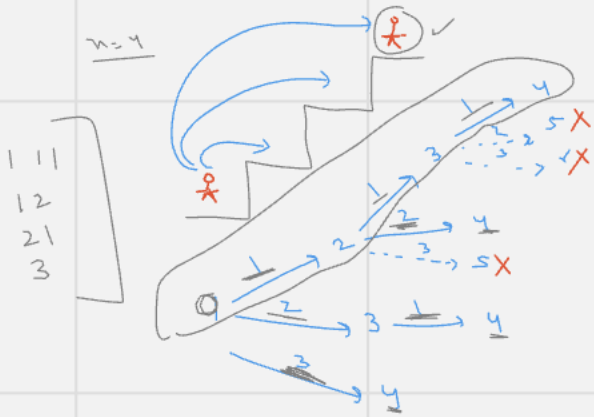
if (0) == (1)

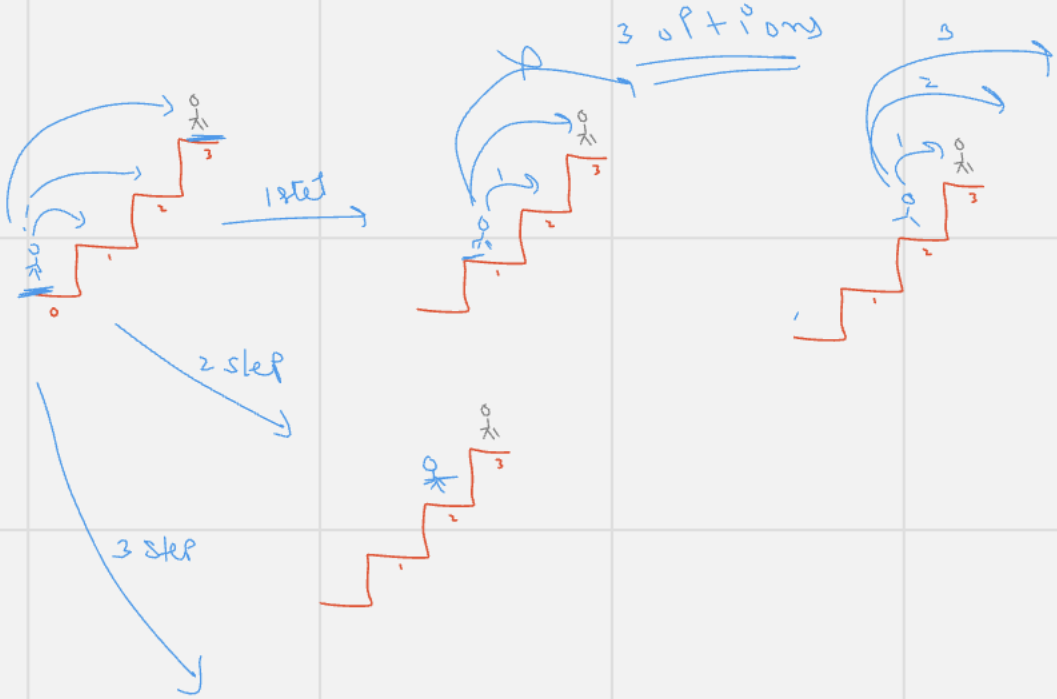
→ h + "*" + recAns

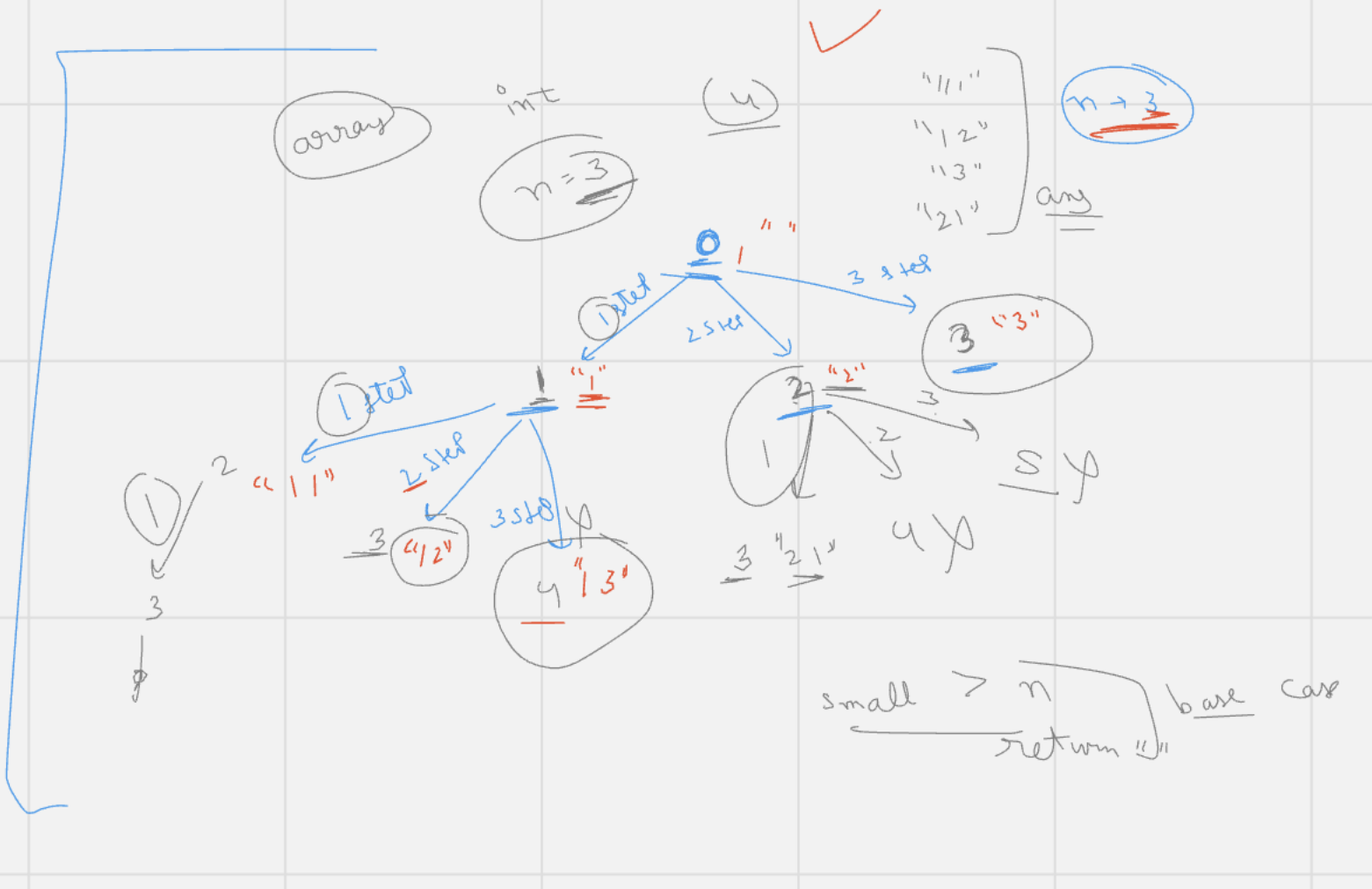
Else

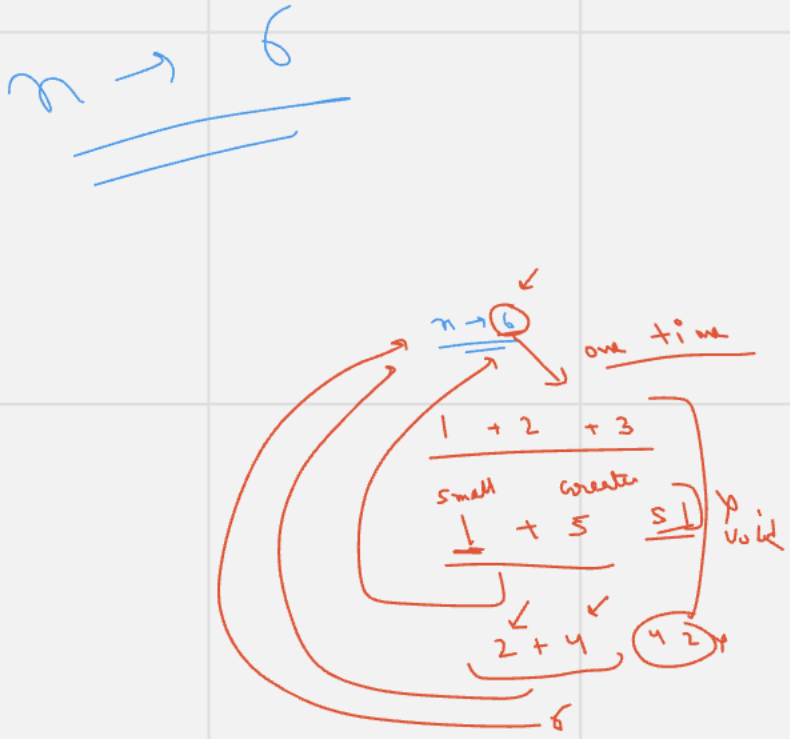
h + recAns


n → 3
1
2
3
1 → 2 → 3
1 → 3


n = 4
1 1 1
1 2
2 1
3


3 options
1 step
2 step
3 step


arrays
int
n = 3
(4)
"111"
"12"
"3"
"21"
any
n → 3
0
" "
1 step
2 step
3 step
1
"1"
2
"2"
3
"3"
1 step
2 step
3 step
1
"11"
2
3
"12"
4
"13"
3
"21"
small > n
base case
return "n"



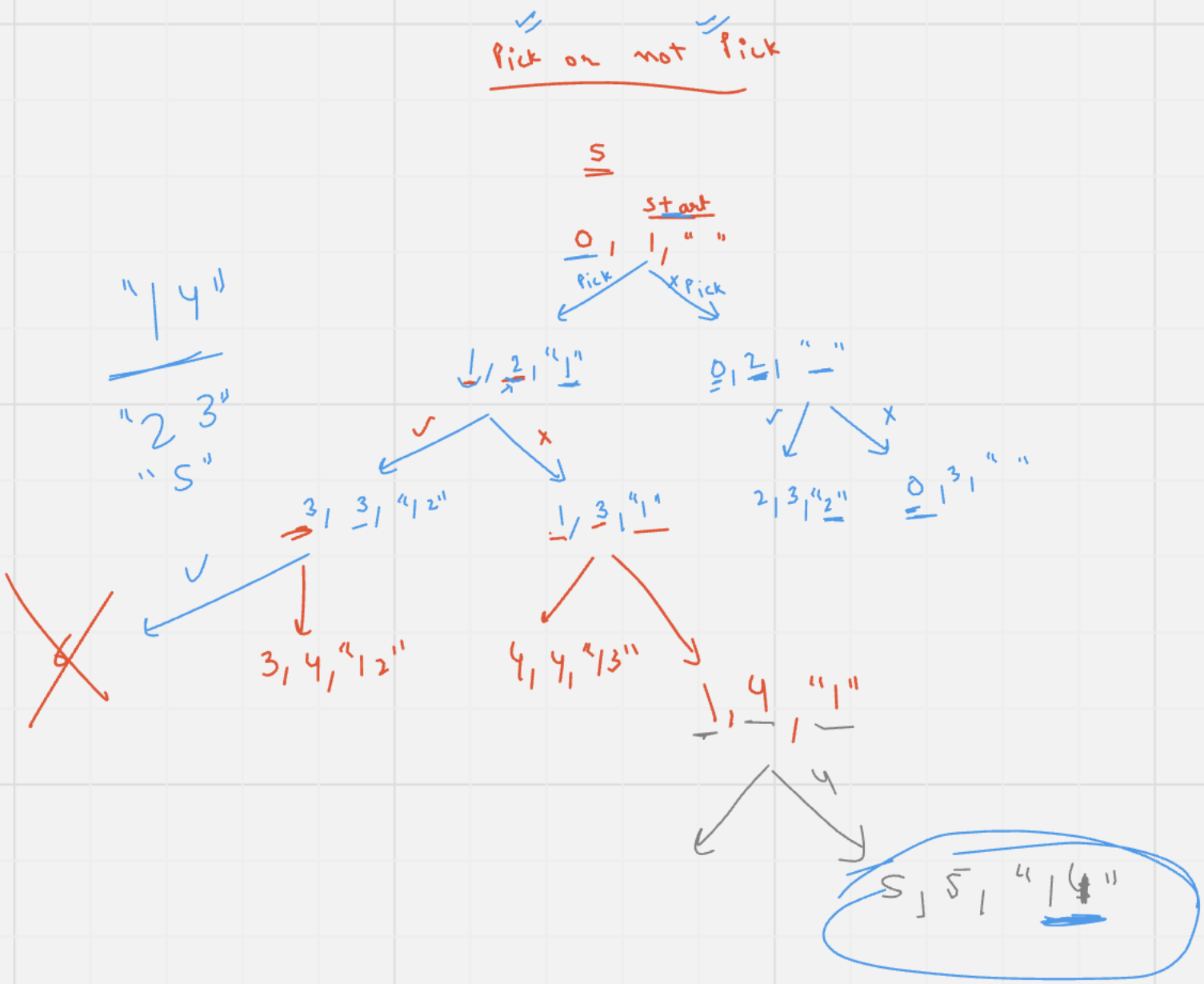
S =

1 4

2 3

5

Ans =


Pick or not Pick
S
start
0 | 1 | " "
Pick
x Pick
"1 4"
"2 3"
"5"
1 | 2 | "1"
0 | 2 | " "
✓
x
✓
x
3 | 3 | "12"
1 | 3 | "1"
2 | 3 | "2"
0 | 3 | " "
✓
3 | 4 | "12"
4 | 4 | "3"
1 | 4 | "1"
✓
5 | 5 | "14"

```
public static long ways(int n, int start, int curr){

    if(curr == n) {
       return 1;
    }

    if(start > n || curr > n) {
       return 0;
    }


    // pick
    long sa1 = ways(n, start + 1, curr + start);

    // not pick
    long sa2 = ways(n, start + 1, curr);

    return sa1 + sa2;

}
```