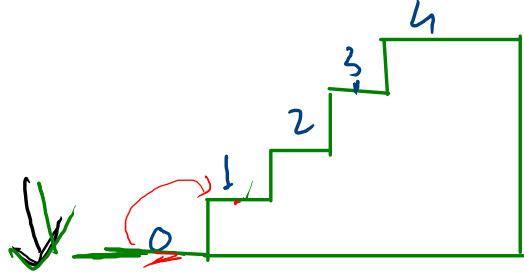
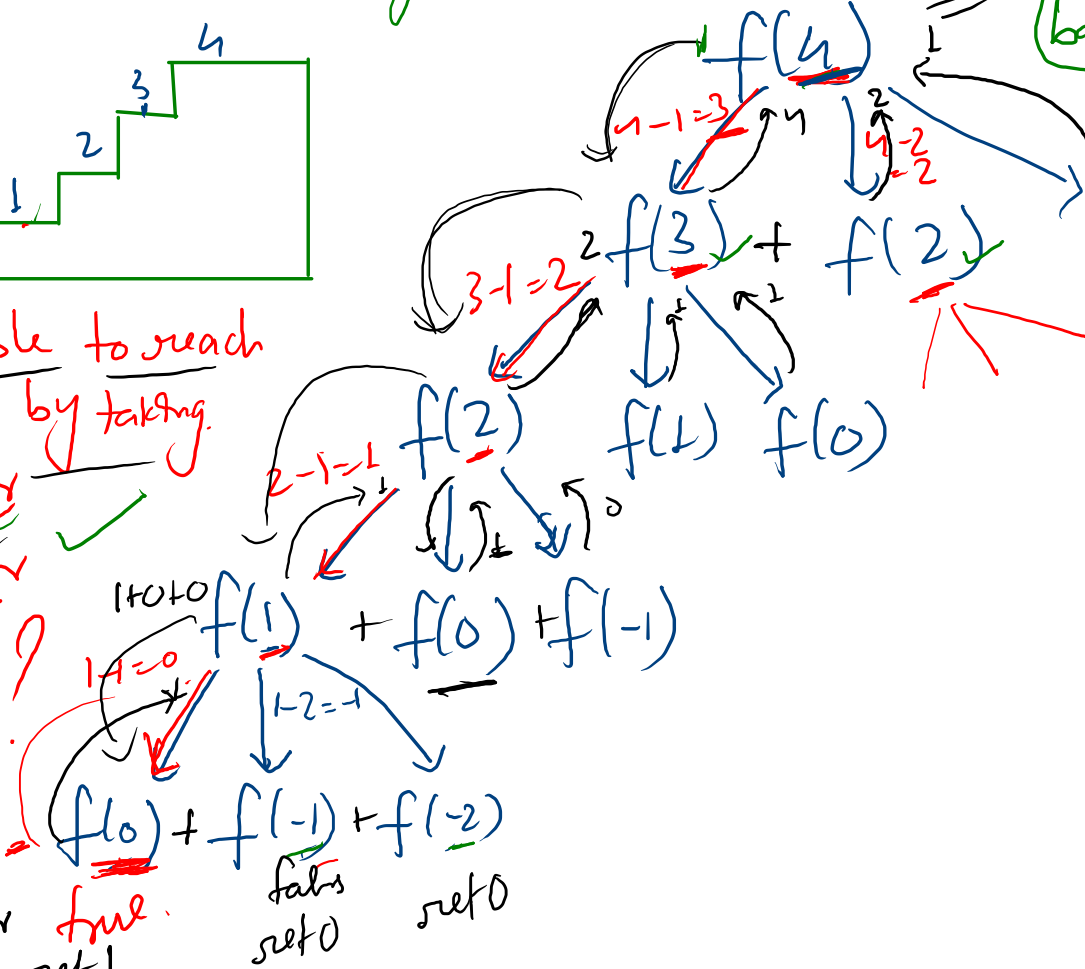


① Number of ways Problem :-



was I able to reach
stairs #n by taking
1 jump or
2 jump or
3 jump?

did this qn
divide problem
into similar
subproblem or
not?



1 jump or 2 jumps or 3 jump.

base case → what is the min
valid value of n
↳ = 0

true

false

set 0

```
→ numOfWays(n)
{
  if (n < 0) return 0
  if (n == 0) return 1

  return (now(n-1) + now(n-2)
          + now(n-3))
}
```



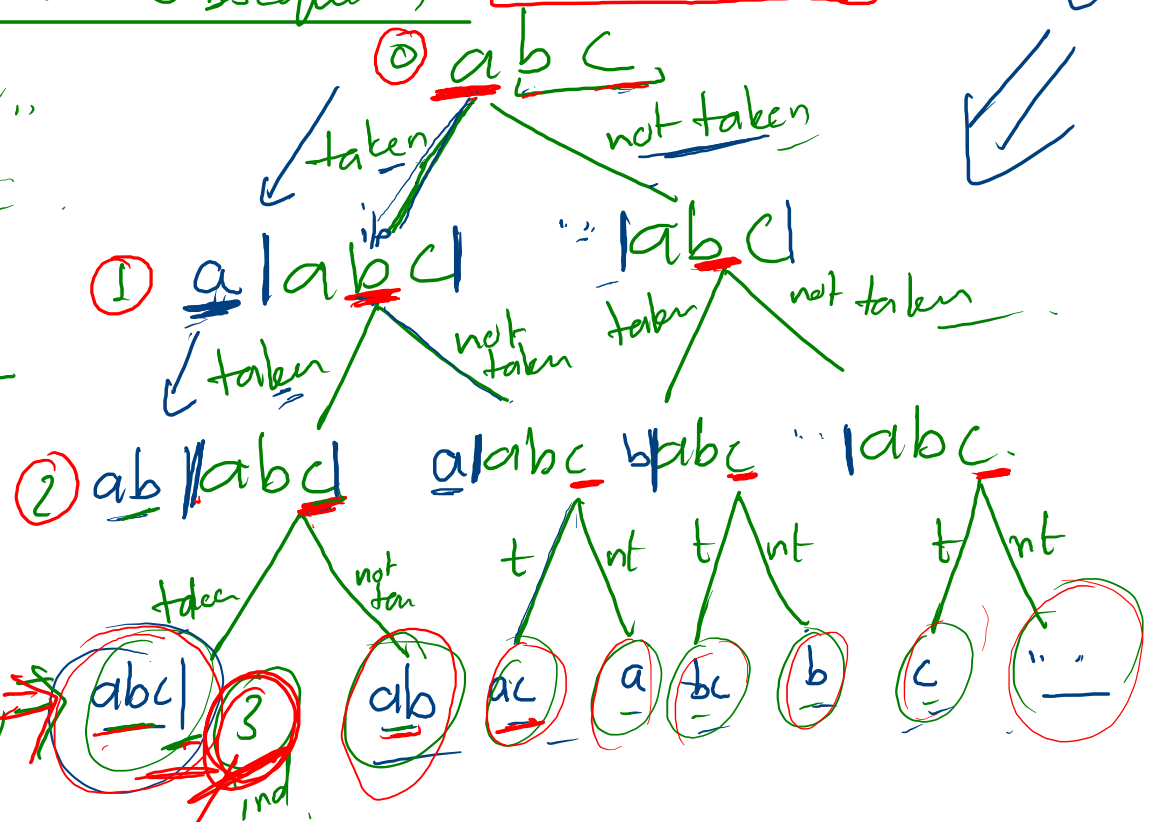
⇒ We need to divide the problem
into similar subproblems
→ ^{then} solⁿ will also be similar

② Generate all subsequences order matters

Choice
Diagram

↓
→ "abc"
substrings
→ a
→ ab
→ abc

→ b
→ bc
→ c
→ ac



ind == str.length
point to stop

do we have any item/char at index 3
in our string!

~~gs(str, ans, ind)~~
~~if (ind >= str.length)~~
~~if (ans.length > 0)~~
~~print ans~~
~~return~~

write extra logic to store

$3 \geq 3 \checkmark$
 $1 > 0 \checkmark$
 $3 \geq 3 \checkmark$
 $0 > 0 \times$

at the end sort the stored subsequences & then print

$\begin{matrix} 0 & 1 & 2 \\ a & b & c \\ \uparrow & \uparrow & \uparrow \\ \text{ind.} & \text{ind} & \text{ind} \end{matrix}$

~~ans = "~~

~~ans += str[ind]~~ ~~faking~~
~~gs(str, ans, ind+1)~~
~~ans.delete(ans.length-1)~~
~~gs(str, ans, ind+1)~~ ← don't take

nothing to execute means automatically return to where it was called

	g4	3
	g3	2
	g2	1
	g1	0
ans.		ind.

fn call stack

print
~~abc~~
~~ab~~
~~ac~~
 a
 bc
 b

str="abc"
↑

abc