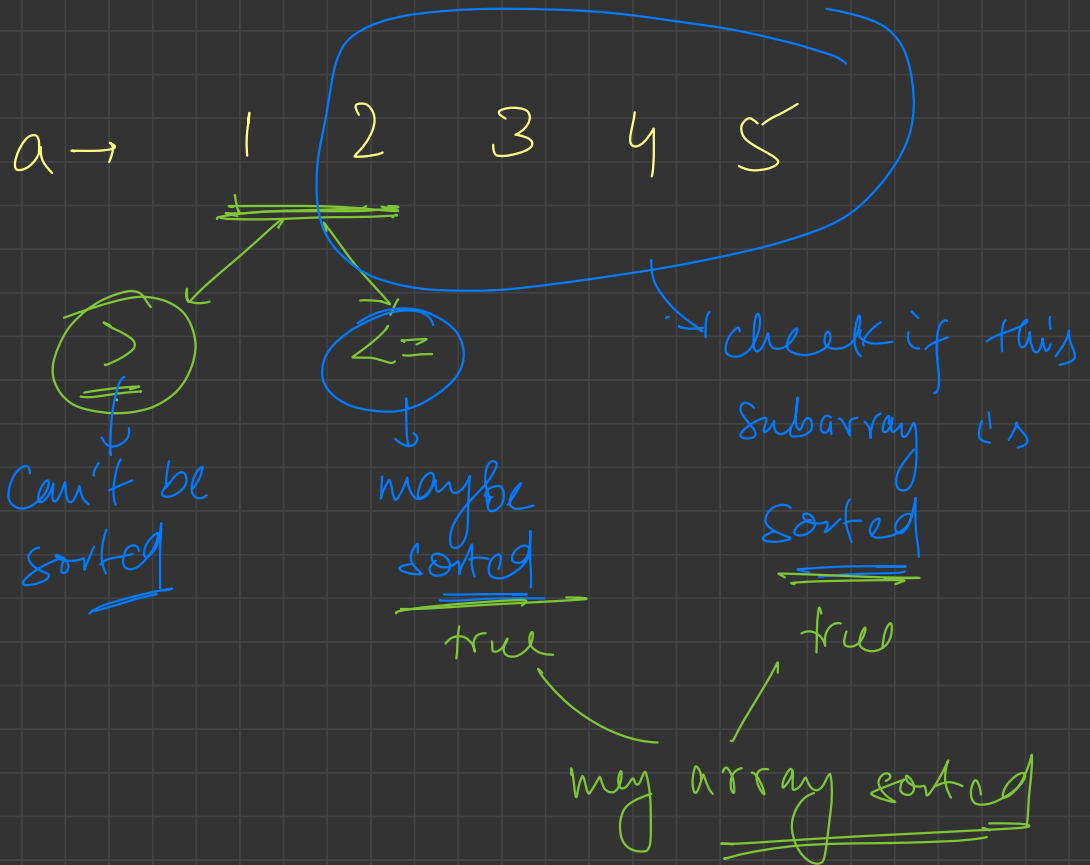
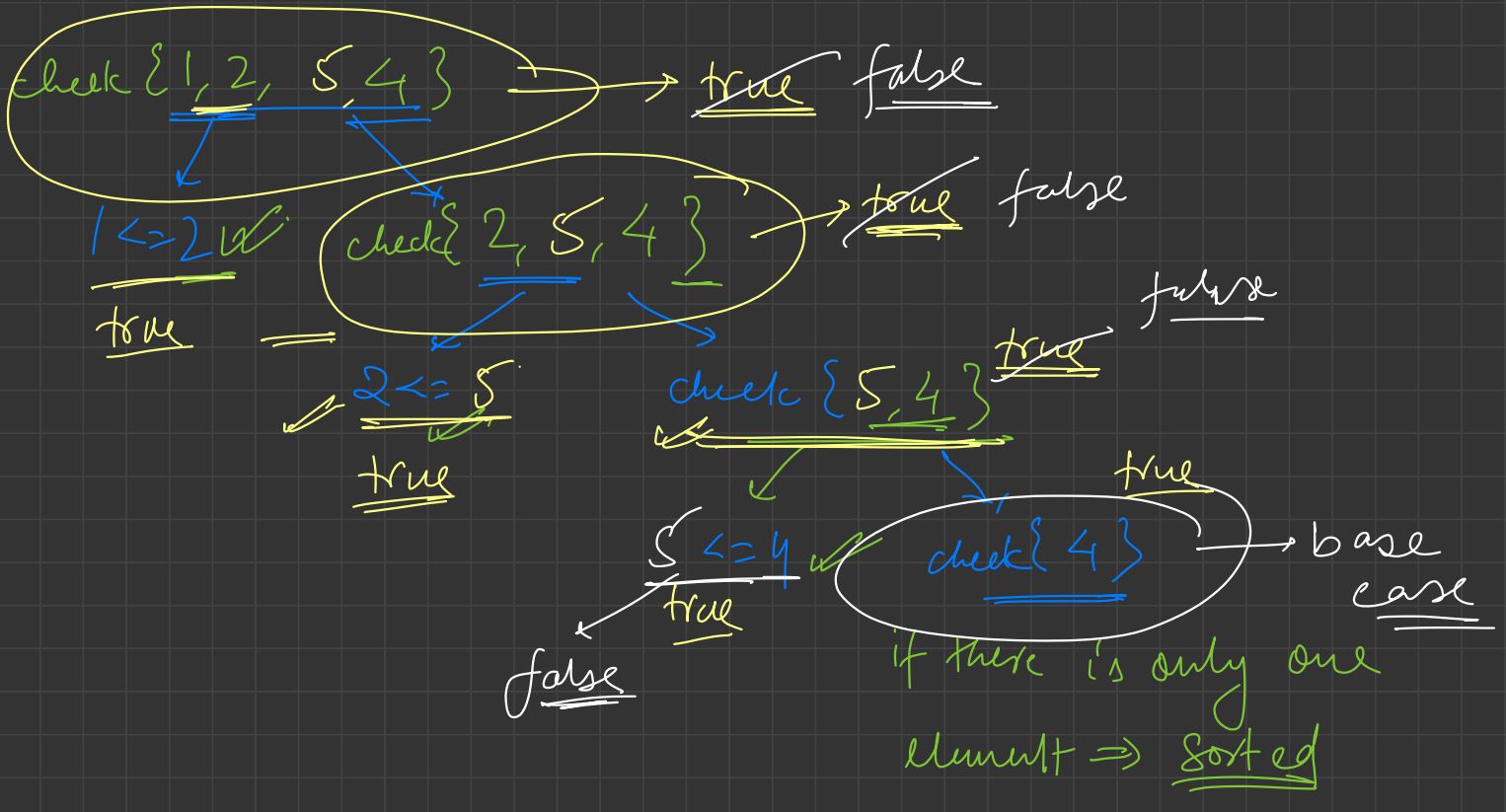
Q. Check whether an array is sorted using recursion?

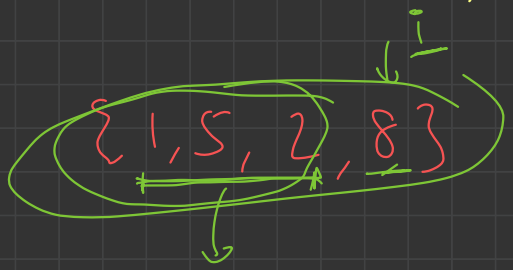



check {1, 2, 5, 4}
true
false
1 <= 2
true
check {2, 5, 4}
true
false
2 <= 5
true
check {5, 4}
true
false
5 <= 4
true
false
check {4}
true
base case
if there is only one element => sorted

```
check(arr, i, n) {
    -- base case
    arr[i] <= arr[i+1] : true
    check(arr, i+1, n) : true
                                  ⇒ true
    otherwise false
}
```

Q. Find sum of array using recursion?

if i want to find the sum → i must go through each element



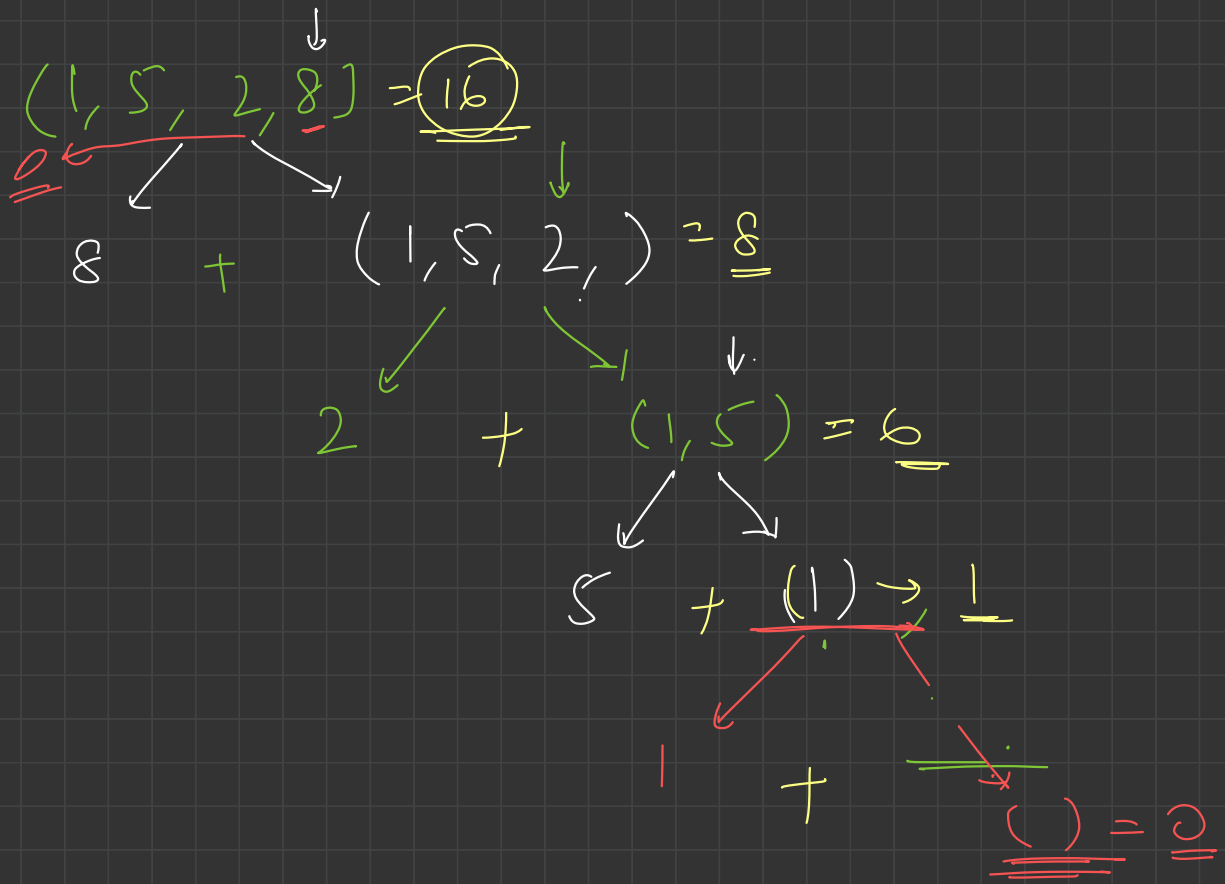
find sum of array upto i-1

+

arr[i]

↓

sum of the array upto i


[1, 5, 2, 8] = 16
8 +
(1, 5, 2,) = 8
2 +
(1, 5) = 6
5 +
(1) → 1
1 +
() = 0

Q: Given an array, and a value (x), search whether it is present in the array or not recursively?

→ find first index of x

→ find last index of x

Break Till 9:45

"1234" → 1234

String to integer

(valid non-negative integer)

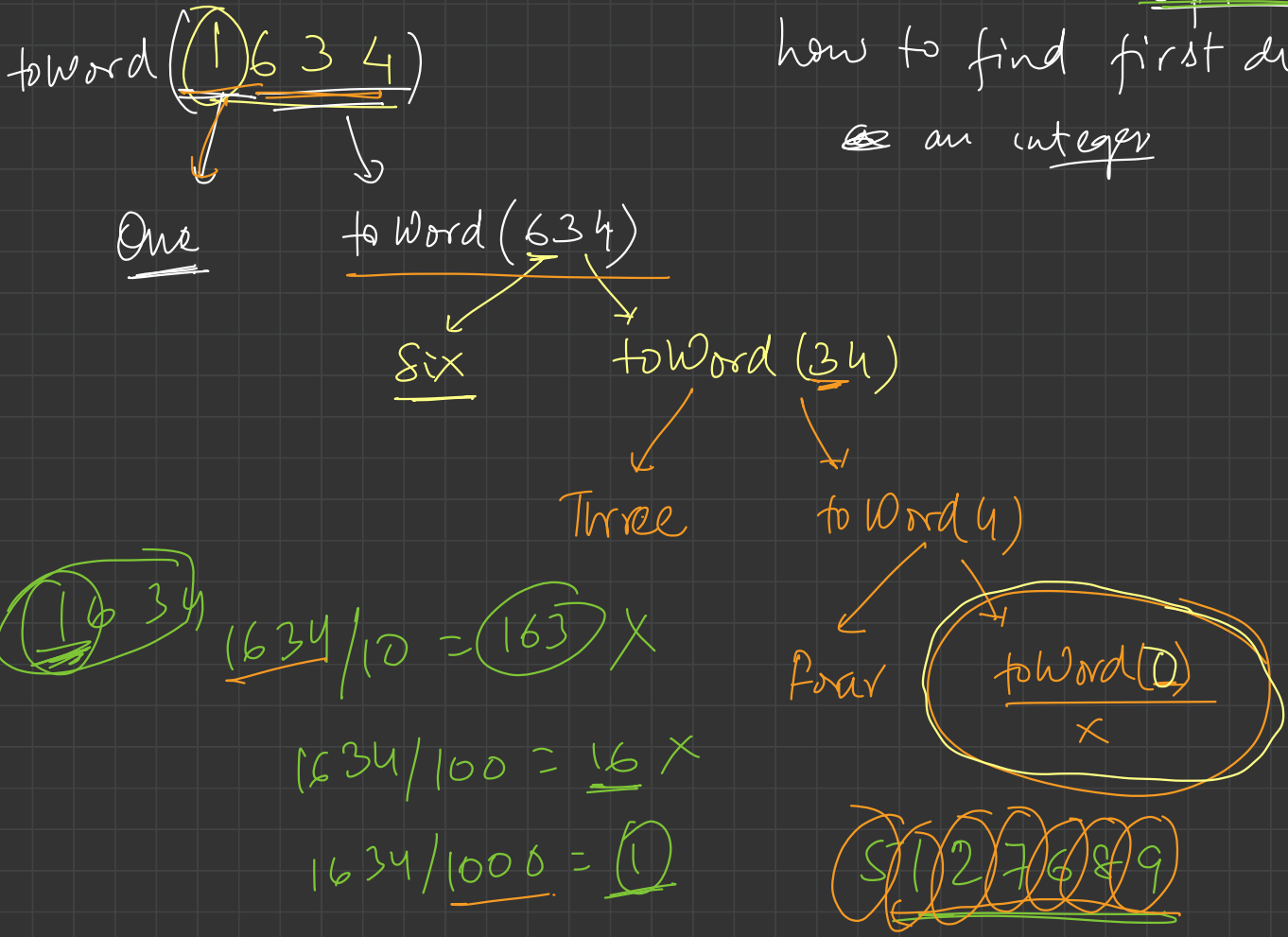
Q. Given a number, convert it into words.

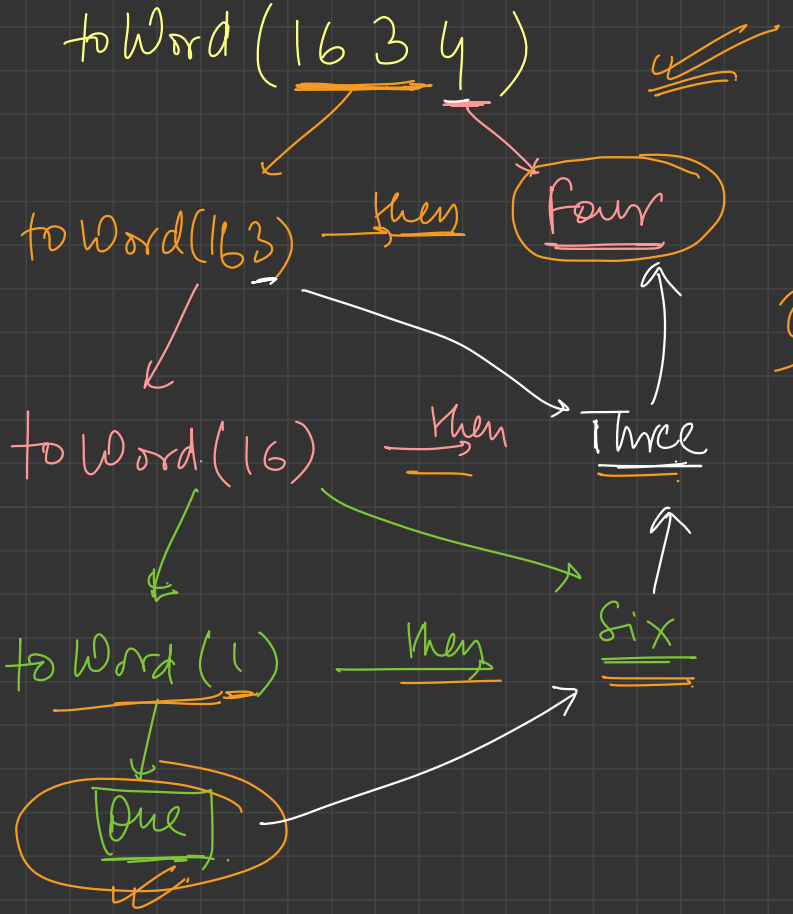
recursively.

163 → One Six Three

Leftmost :-

how to find first digit of an integer



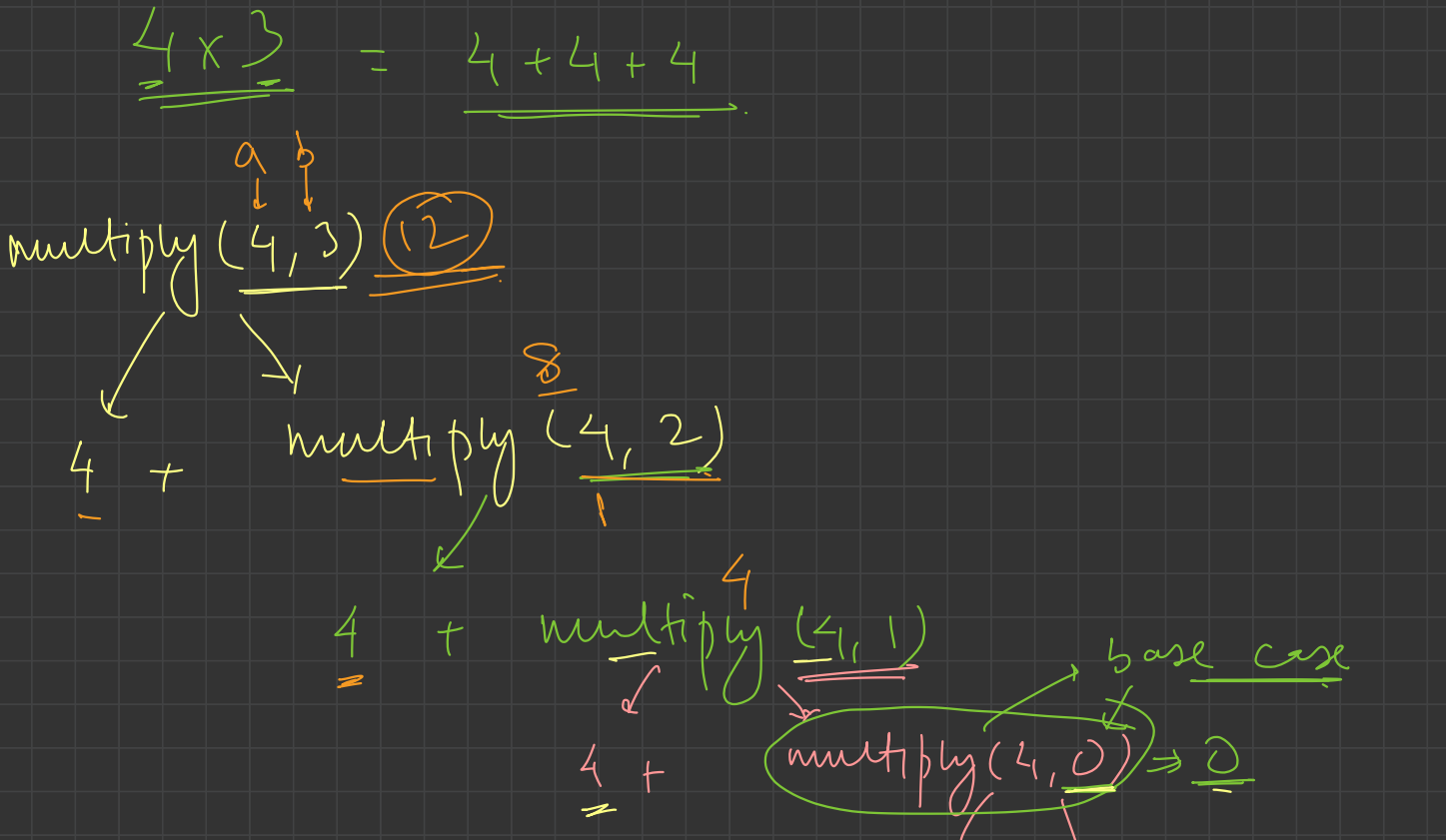


%10 ⇒ last digit

Q. how to find length/no. of digits in an int

```
cn=0
while(n>0){
    cn++;
    n/=10;
}
```

Q. Multiply two numbers using recursion ?



How to optimize no. of recursive calls

② if only a is negative

③ if only b is negative

④ if both are negative

Q- Calculate pow(a, b) using recursion!!

pow(2, 3) = 8

$2^3 = 8$