

function call

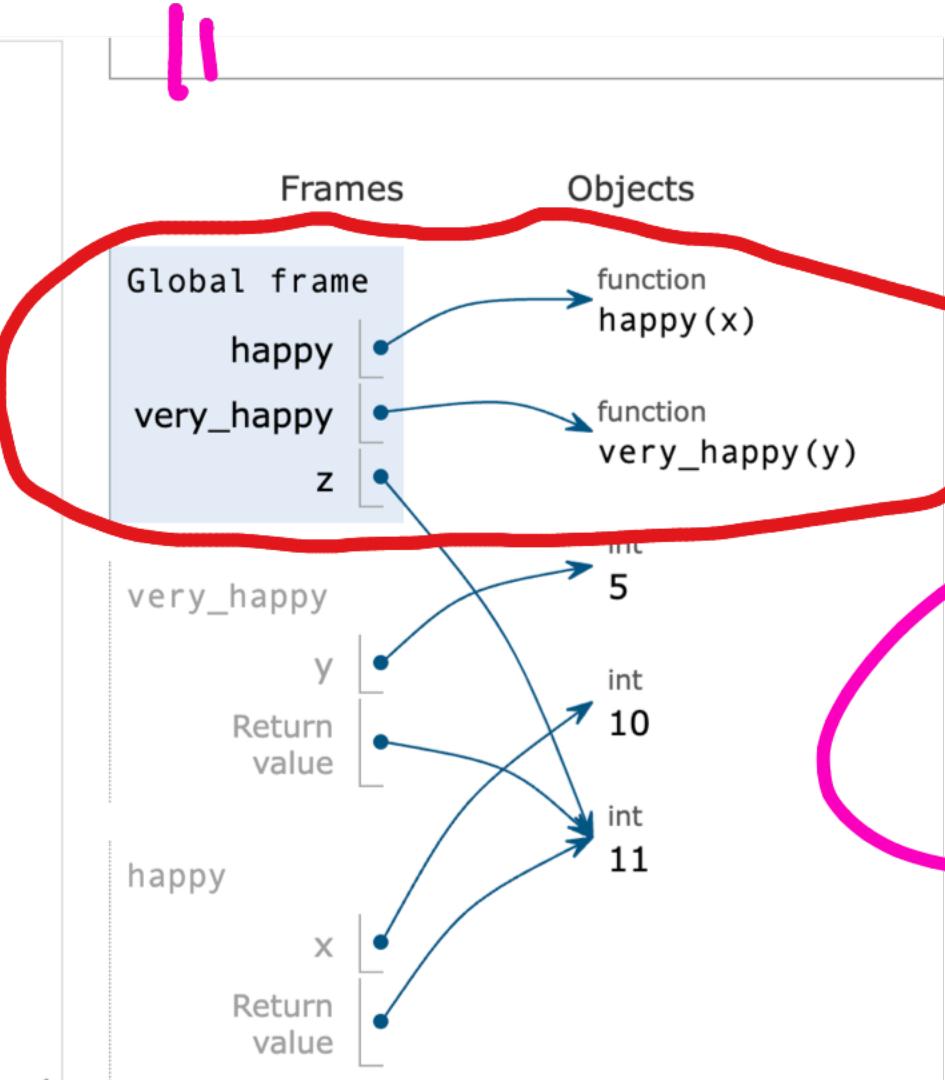
Function Calling

LIFO

```

1 def happy(x):
2     return x+1
3
4 def very_happy(y):
5     return happy(2*y)
6
7 z = very_happy(5)
8 print(z)
9
10
11

```



destroyed

in happy(10)
very_happy(5)

11
Stack is used in function calling

Recursion

```
In [25]: def happy(x):  
    if x == 5:  
        return x  
    return happy(x+1)
```

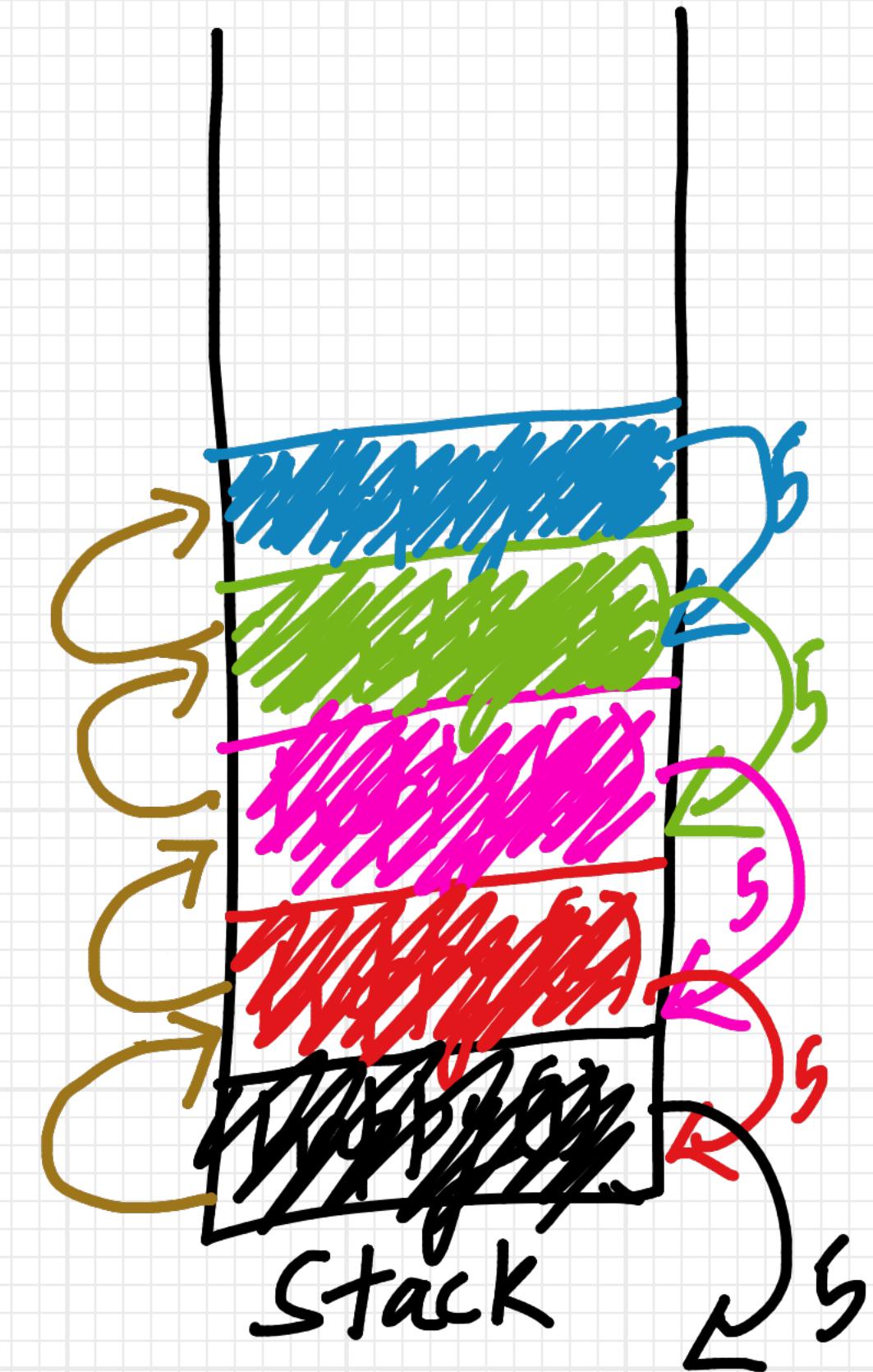
```
In [26]: y = happy(1)
```

```
In [27]: print(y)
```

5 ✓

function
function call

happy function call
is calling with
function $x+1$ value



Recursion

In [25]:

```
def happy(x):
    if x == 5:
        return x
    return happy(x+1)
```

In [26]:

```
y = happy(1)
```

In [27]:

```
print(y)
```

5

Never gets paused →
pause ↓
play pause 11
play pause 11
play pause 11
play pause 11
play



Function Calling

```
In [10]: def happy(x):  
         return x+1
```

```
In [11]: def very_happy(y):  
         return happy(2*y)
```

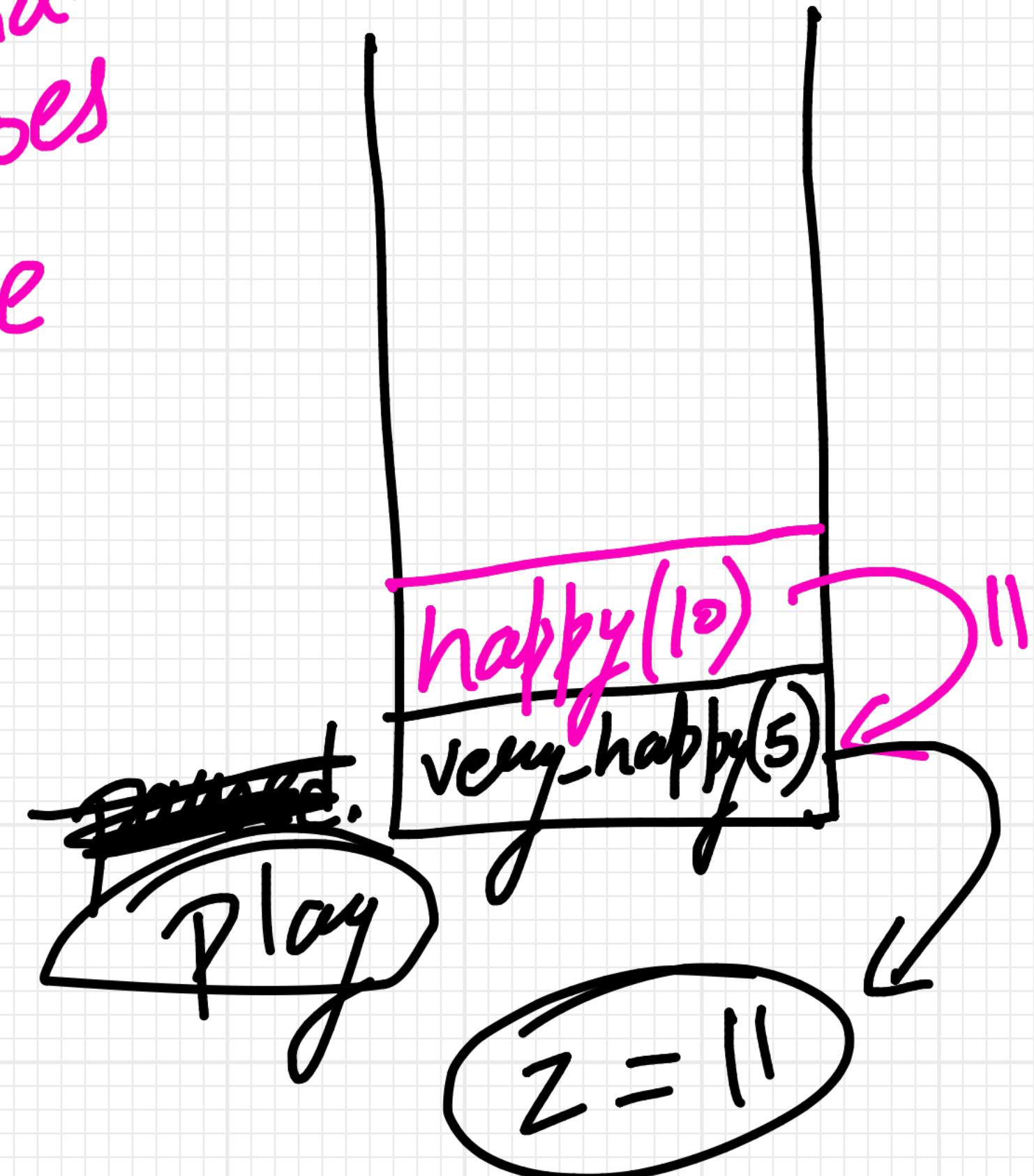
```
In [13]: z = very_happy(5)
```

```
In [14]: print(z)
```

11



When it calls another function,
it itself goes
into pause
mode



Quick Sort

QuickSort(A, l, r)

{ if $l \geq r$:
 return }

$m \leftarrow \text{Partition}(A, l, r)$

{ $A[m]$ is in the final position }

QuickSort($A, l, m - 1$)

QuickSort($A, m + 1, r$)

• Recursion

leftmost index

sightmost index

Considered hist

5	-3	33	29	16
---	----	----	----	----

pivot

-3	5	16	29	33
----	---	----	----	----

$l = u$

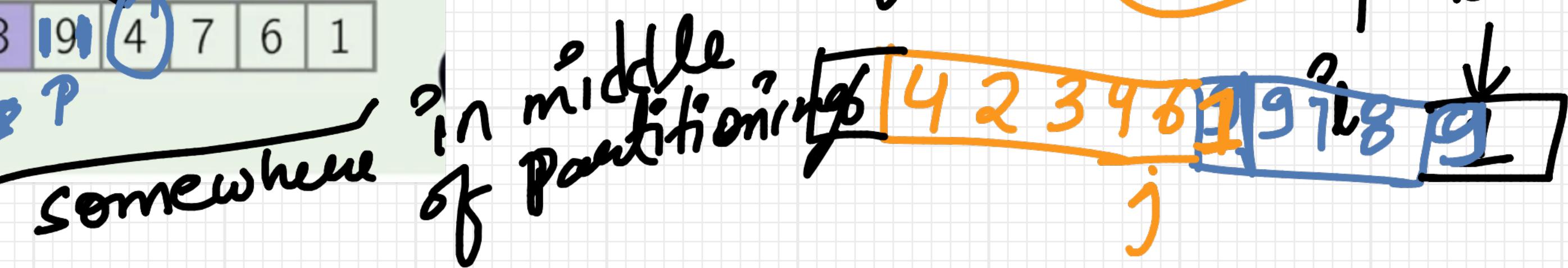
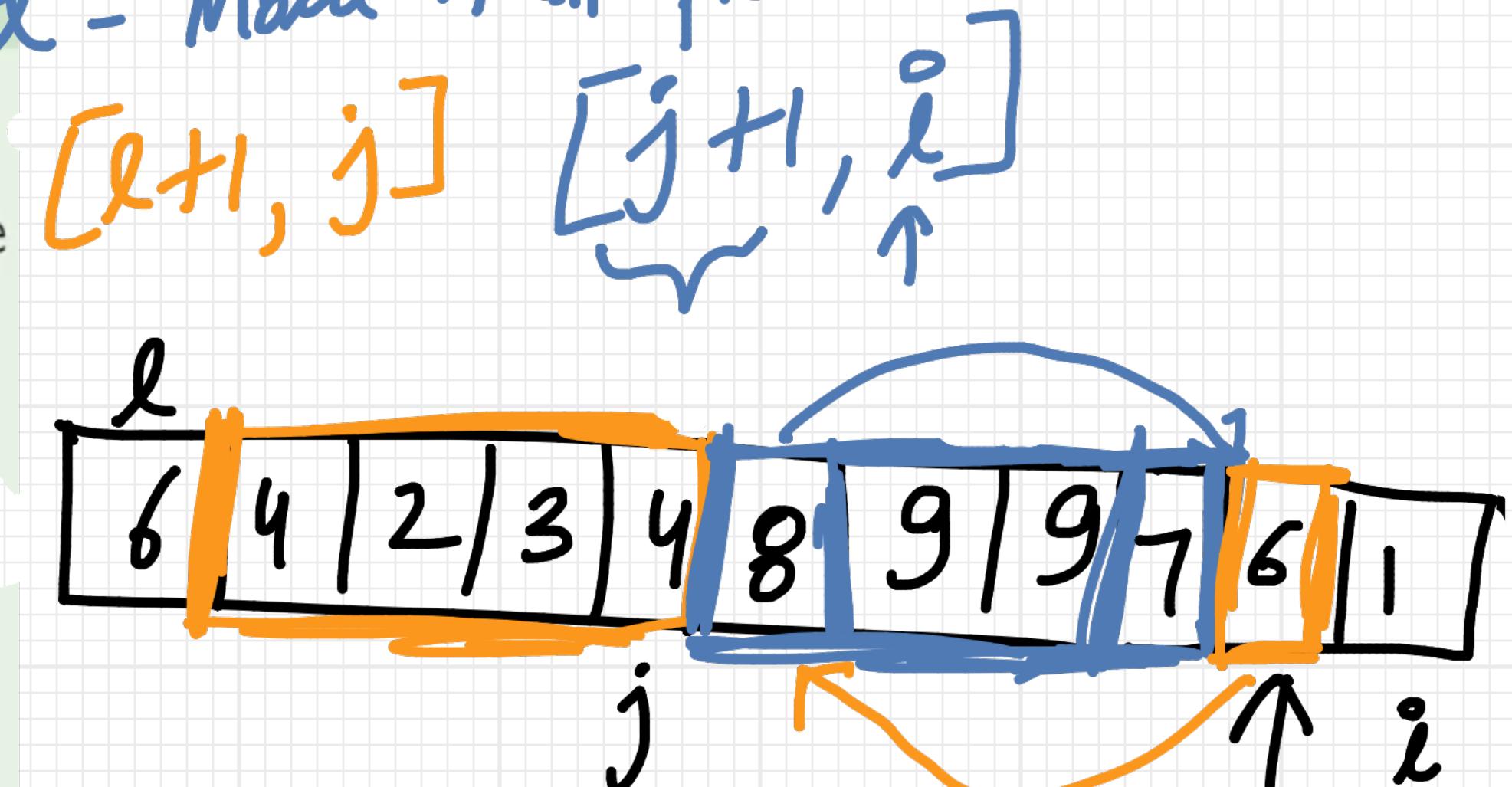
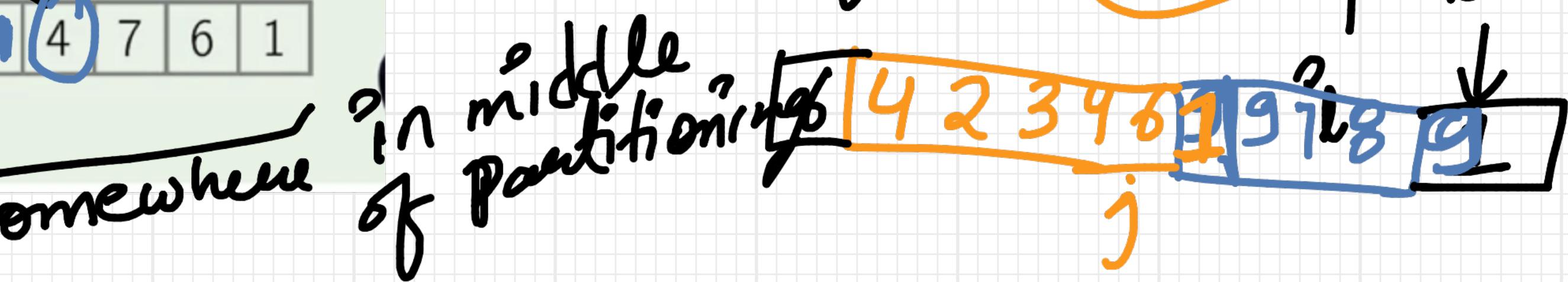
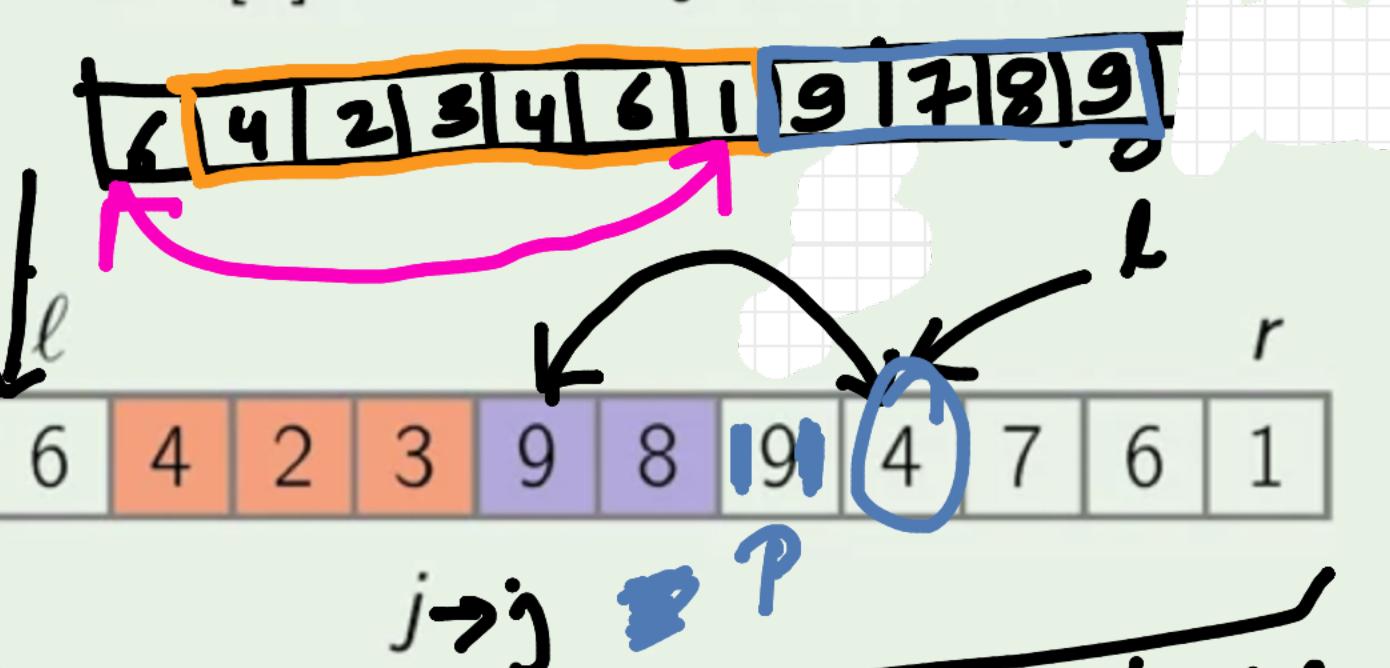
$l > r$

Partitioning = figuring out the final index position of the pivot element, once it is sorted.

Partitioning: example

- the pivot is $x = A[\ell]$
- move i from $\ell + 1$ to r maintaining the following invariant:

- $A[k] \leq x$ for all $\ell + 1 \leq k \leq j$
- $A[k] > x$ for all $j + 1 \leq k \leq i$



Partitioning: example

- the pivot is $x = A[\ell]$
- move i from $\ell + 1$ to r maintaining the following invariant:
 - $A[k] \leq x$ for all $\ell + 1 \leq k \leq j$
 - $A[k] > x$ for all $j + 1 \leq k \leq i$
- in the end, move $A[\ell]$ to its final place

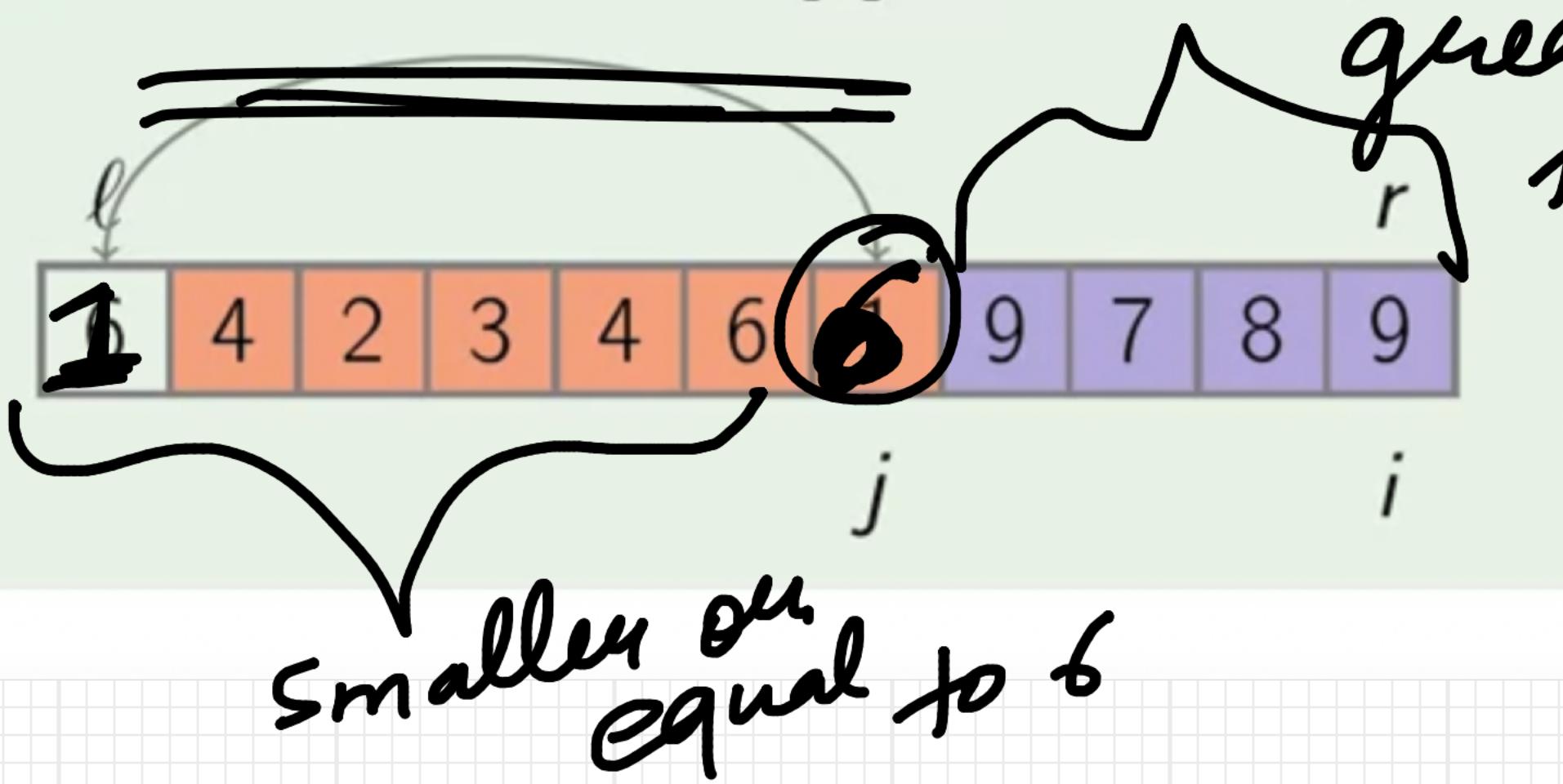


Diagram illustrating the partitioning step of quicksort:

- The array is partitioned around the pivot value 6.
- The region to the left of the pivot ($\ell + 1$ to i) contains elements less than or equal to 6, labeled "smaller or equal to 6".
- The region to the right of the pivot ($j + 1$ to r) contains elements greater than 6, labeled "greater than 6".
- The pivot value 6 is moved to its final place at index i .
- The condition "for every element if above holds" ensures the correctness of the partitioning.
- The final state is labeled "sorted" with a checkmark.