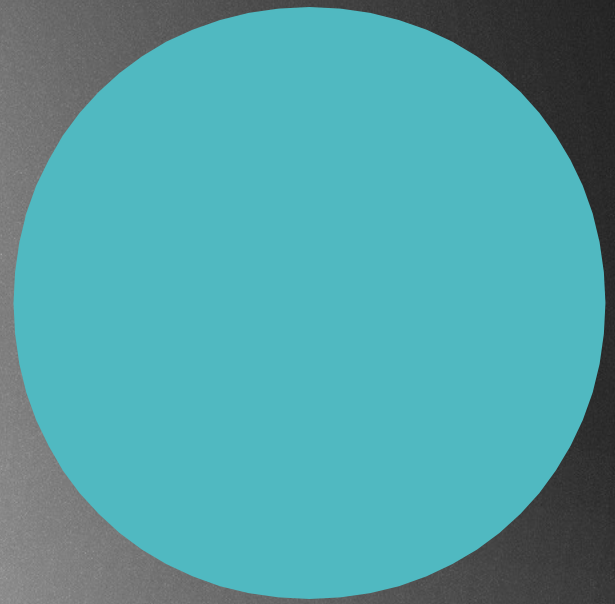
SORTING ALGORITHMS



QUICKSORT

Quicksort

- It is an efficient sorting algorithm
- It was developed by Tony Hoare in 1959 // the same person who invented quickselect algorithm
- A well implemented quicksort can outperforms heapsort and merge sort → the main competitors of quicksort
- A comparison based algorithm → not able to be faster than linearithmic time complexity

Quicksort

- The efficient implementation of quicksort is **NOT** stable → does not keep the relative order of items with equal value
- It is in-place → does not need any additional memory
- On average case it has **O(N logN)** running time
- But the worst case running time is quadratic **O(N^2)** !!!
- It is widely used in programming languages

  Primitive types → usually quicksort is used

  Reference types / objects → usually mergesort is used

Quicksort

It is a **divide and conquer** algorithm

- pick an element from the array: this is the pivot item
- partition phase: rearrange the array so that all elements with values less than the pivot come before the pivot, while all elements with values greater than the pivot come after it // equal values can go either way
- recursively apply these steps on the subarrays

BASE CASE FOR RECURSION: arrays of size zero or one never need to be sorted

Choosing the **pivot item**

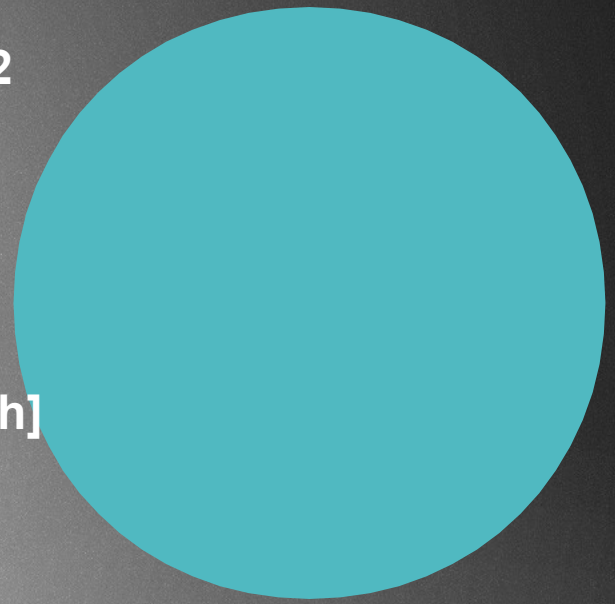
It is very important → if we keep choosing bad pivots, the running time will be quadratic

1.) we can choose a pivot at random // usually it is working fine
2.) choose the middle index of the array as the pivot

Pseudocode

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

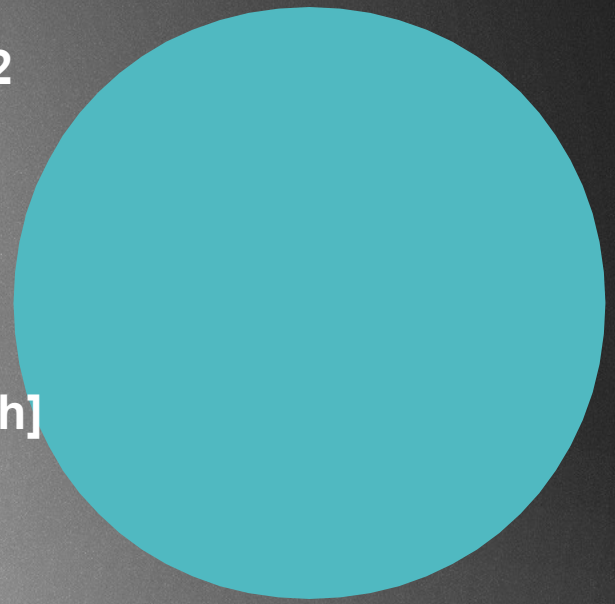
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



Pseudocode

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

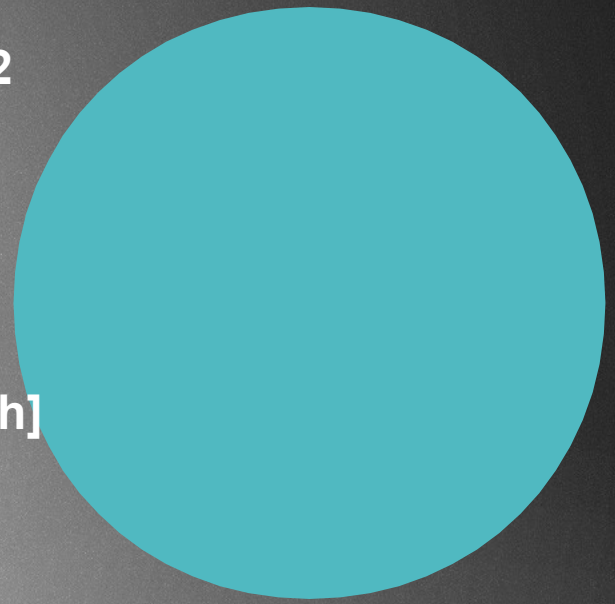


This is the base case for recursion: to track when to terminate our algoithm !!!

Pseudocode

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

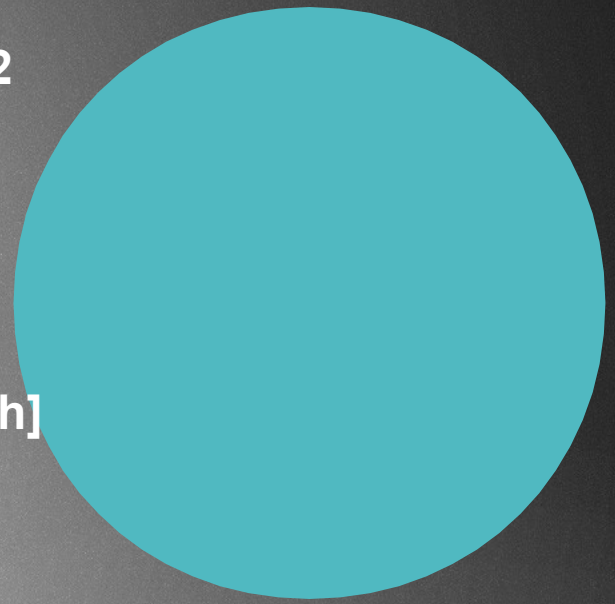


We have to find a pivot item → we are going to partition the array according to this item !!!

Pseudocode

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

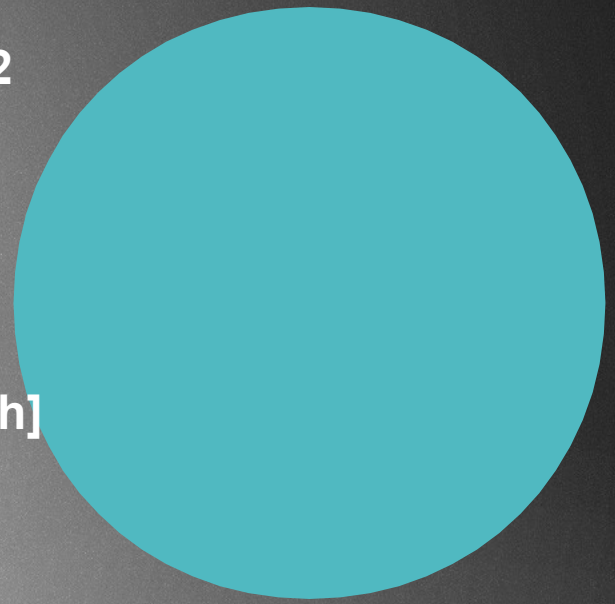


We call the sort method recursively on the left subarray

Pseudocode

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

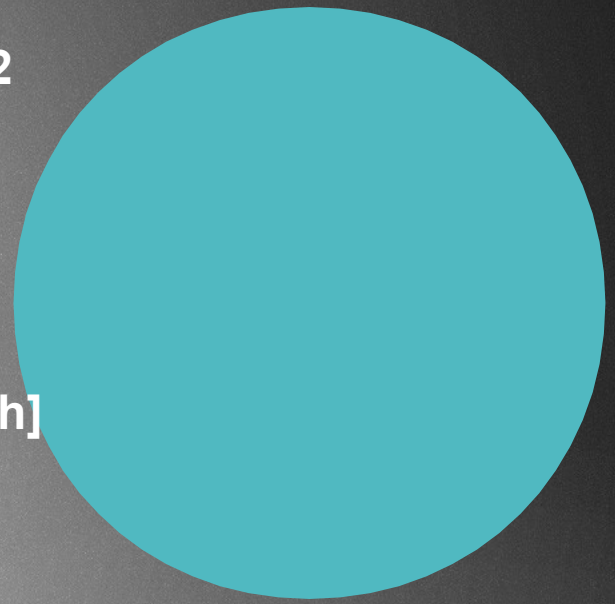


We call the sort method recursively on the right subarray

Pseudocode

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

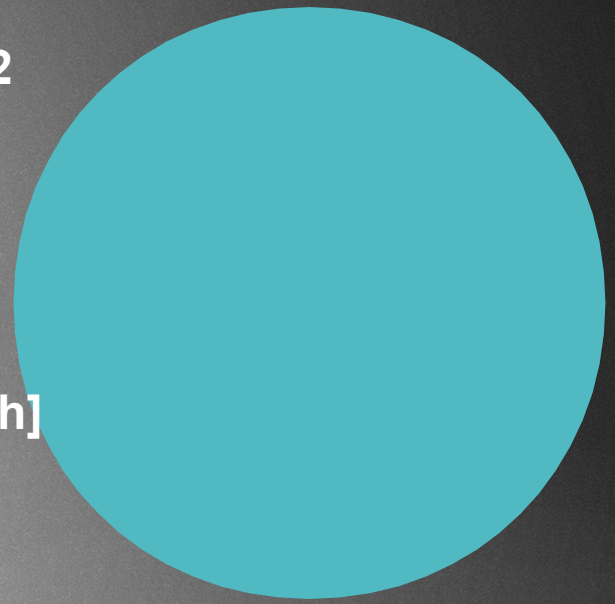


We should generate a random index for the pivot … OR we can have the middle item of the array as the pivot !!!

Pseudocode

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

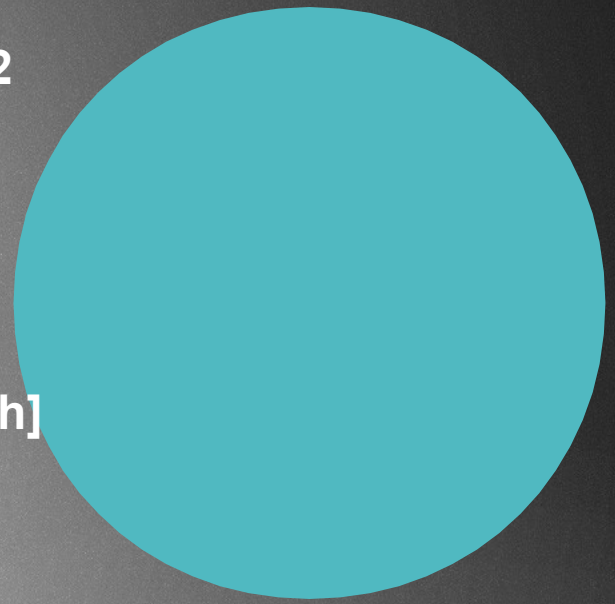


We make the pivot to be the last item in the array !!!

Pseudocode

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

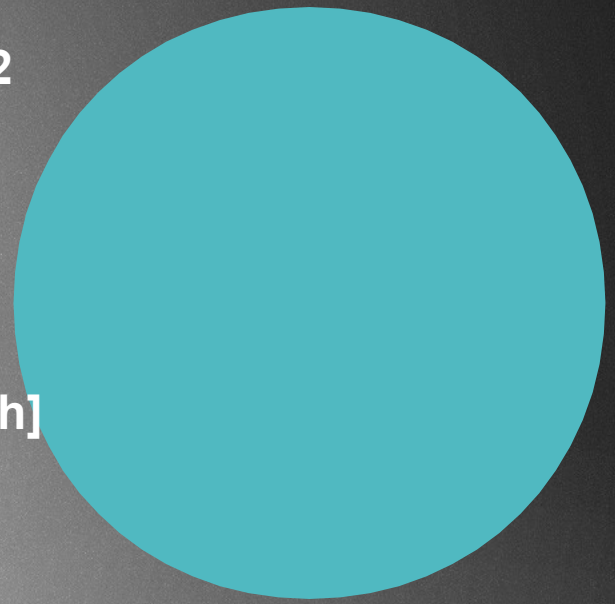


This loop will decide what are the items that will be on the left side of the pivot or the right side of the pivot !!!

Pseudocode

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

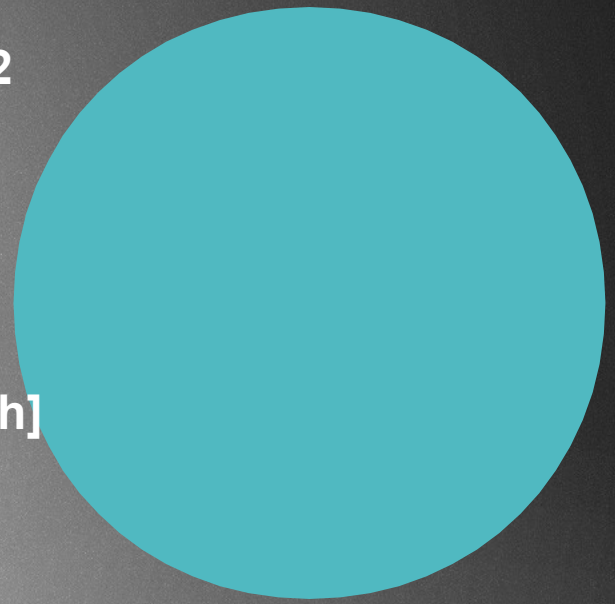


Variable **i** tracks the threshold item → separates items that are smaller / greater than the pivot !!!

Pseudocode

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



We return the index of the pivot so the threshold index !!!

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

23	6	4	-1	0	12	8	3	1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

23	6	4	-1	0	12	8	3	1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

23	6	4	-1	0	12	8	3	1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

23	6	4	-1	1	12	8	3	0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

23	6	4	-1	1	12	8	3	0

i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

23	6	4	-1	1	12	8	3	0

i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

23	6	4	-1	1	12	8	3	0

i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

23	6	4	-1	1	12	8	3	0

i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

23	6	4	-1	1	12	8	3	0

i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

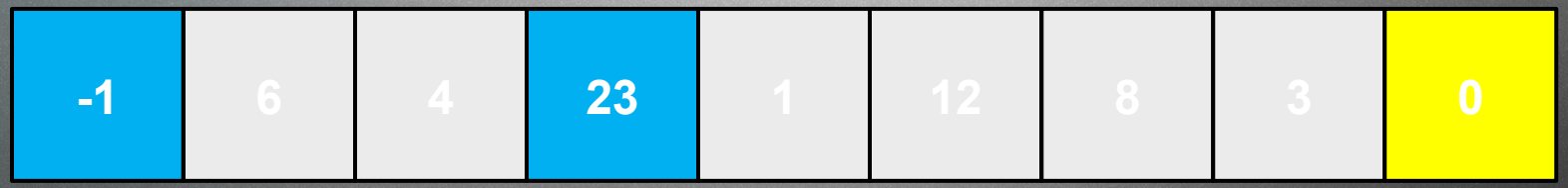
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

23	6	4	-1	1	12	8	3	0

i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	6	4	23	1	12	8	3	0

i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	6	4	23	1	12	8	3	0

i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	6	4	23	1	12	8	3	0

i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	6	4	23	1	12	8	3	0

i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	6	4	23	1	12	8	3	0

i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	6	4	23	1	12	8	3	0

i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	6	4	23	1	12	8	3	0

i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	6	4	23	1	12	8	3	0

i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	23	1	12	8	3	6

i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

So: we have managed to rearrange the array so that on the left side of the pivot → items that are smaller and on the right are the greater items

-1	0	4	23	1	12	8	3	6

i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

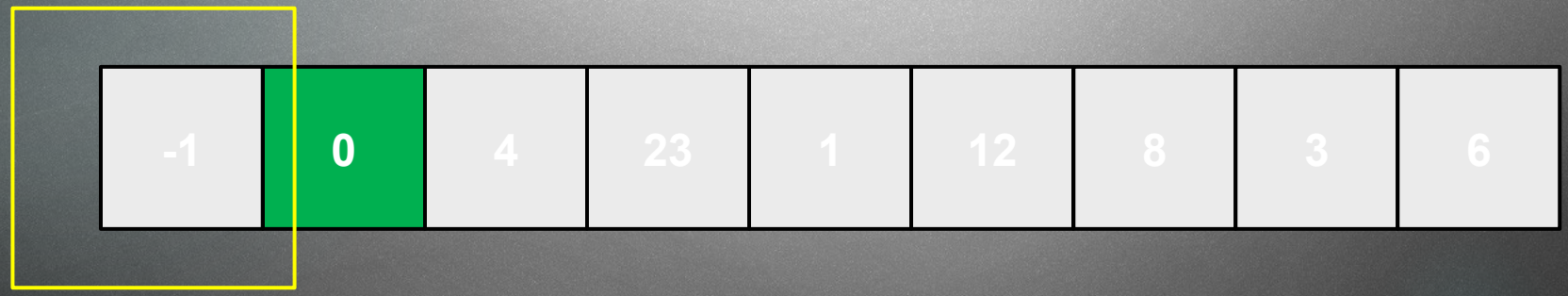
AND we know for sure this item is in it's final position !!!

-1	0	4	23	1	12	8	3	6

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

Sort left subarray recursively



```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	23	1	12	8	3	6

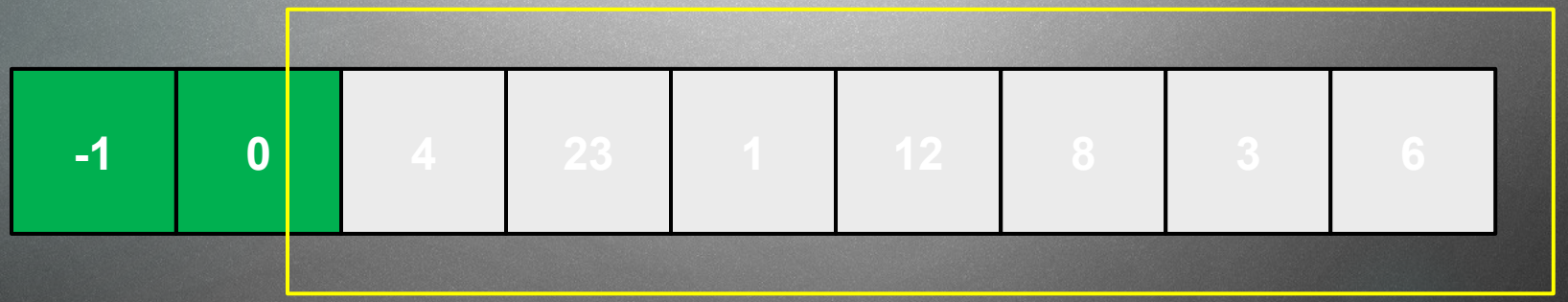
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	23	1	12	8	3	6

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



Sort right subarray recursively

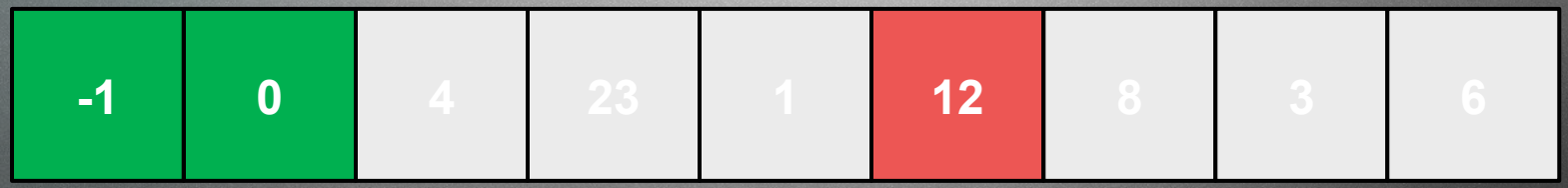
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	23	1	12	8	3	6

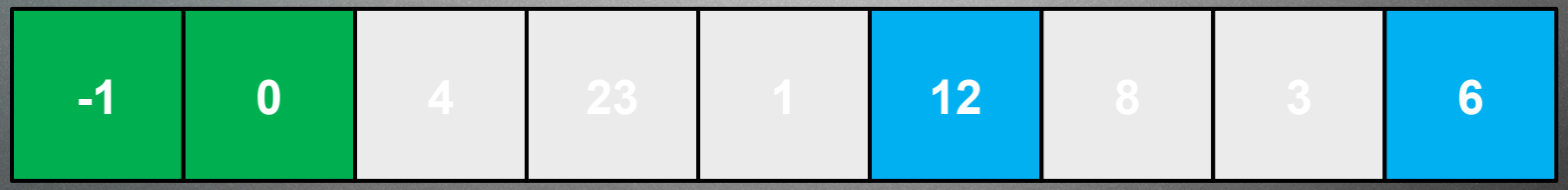
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



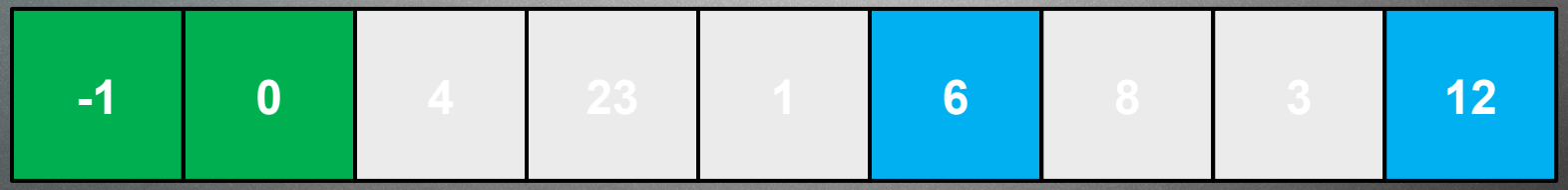
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



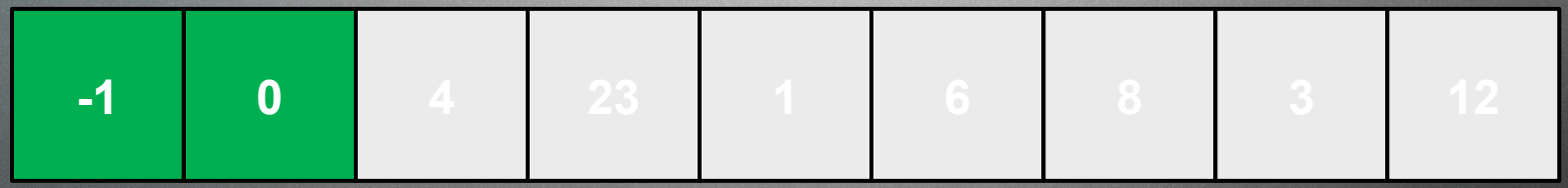
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	23	1	6	8	3	12

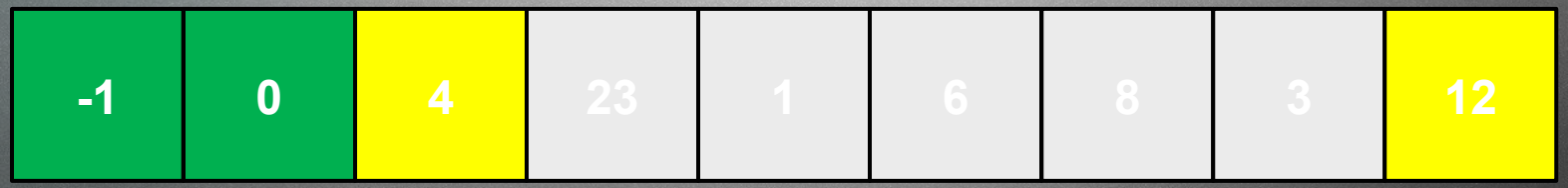
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



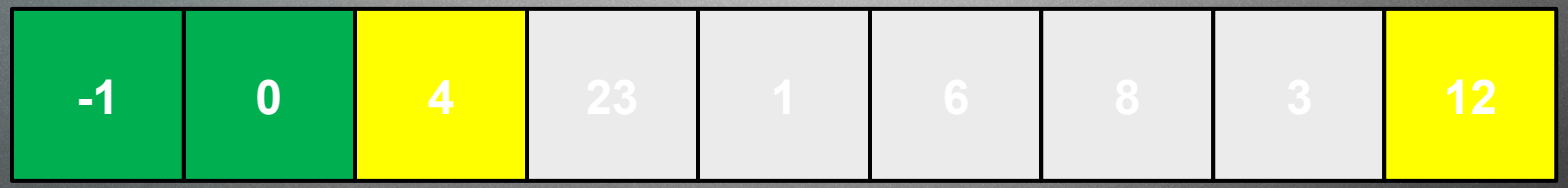
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



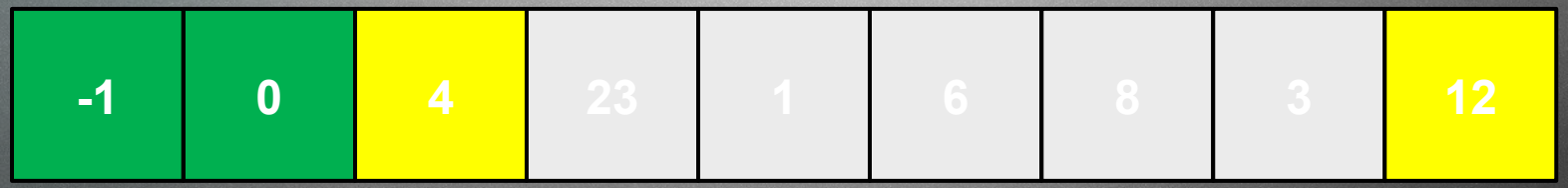
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



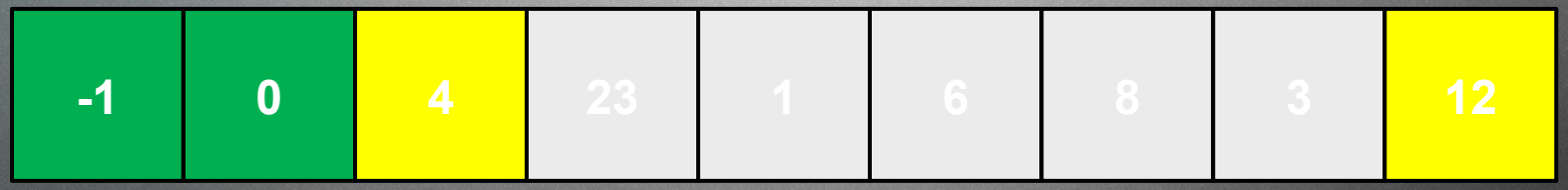
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

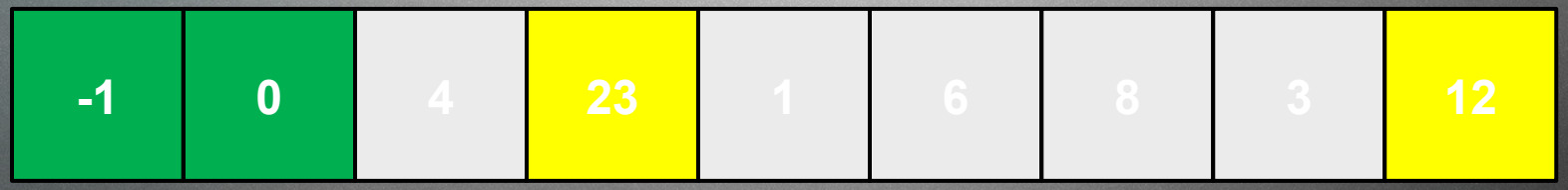
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	23	1	6	8	3	12

i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

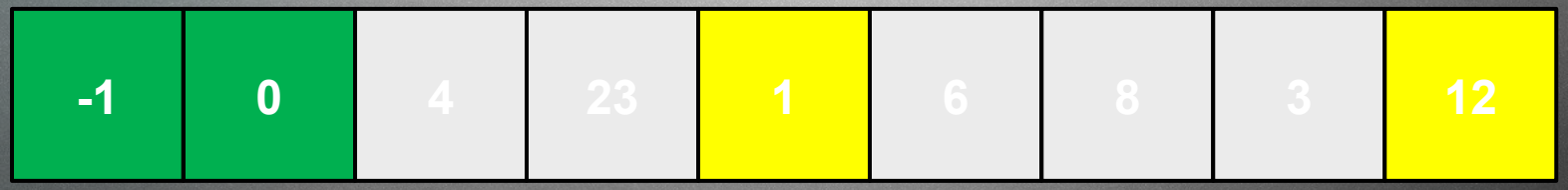
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

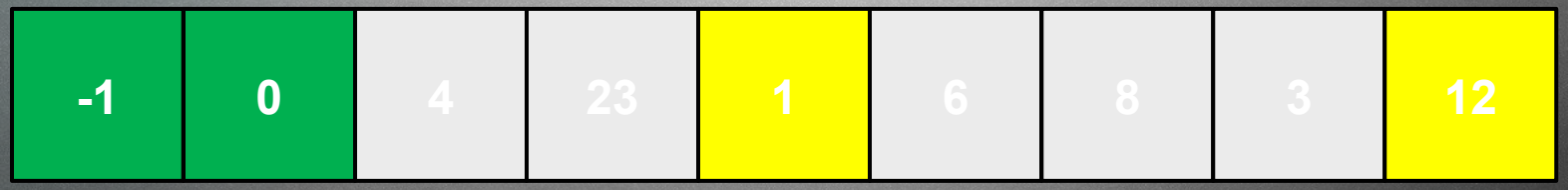
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

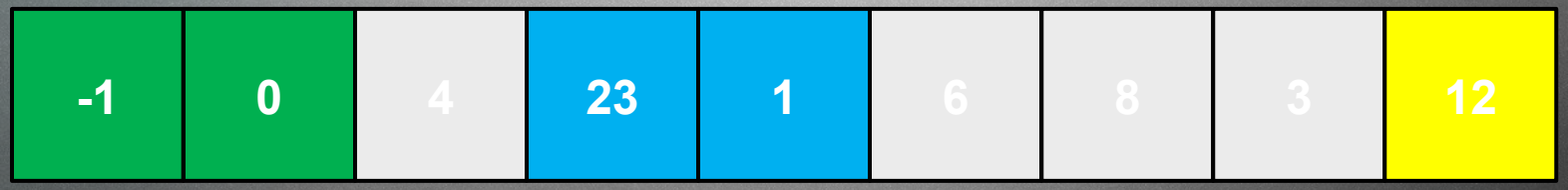
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

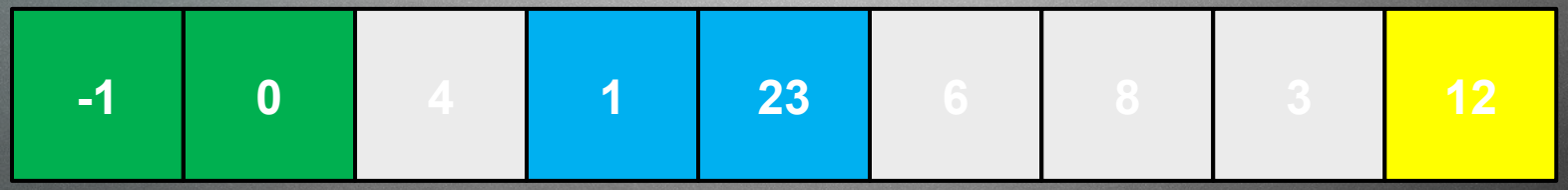
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

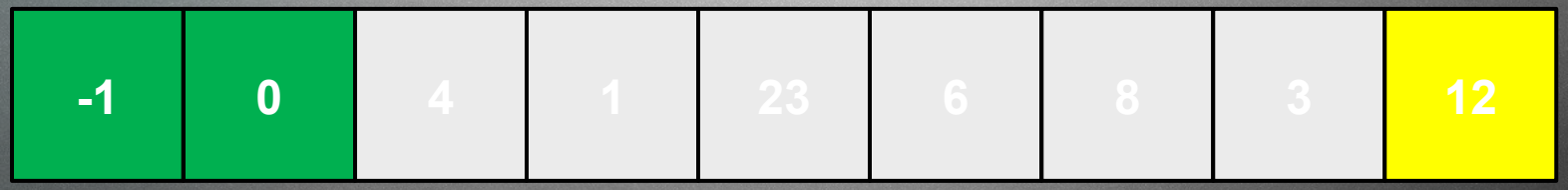
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

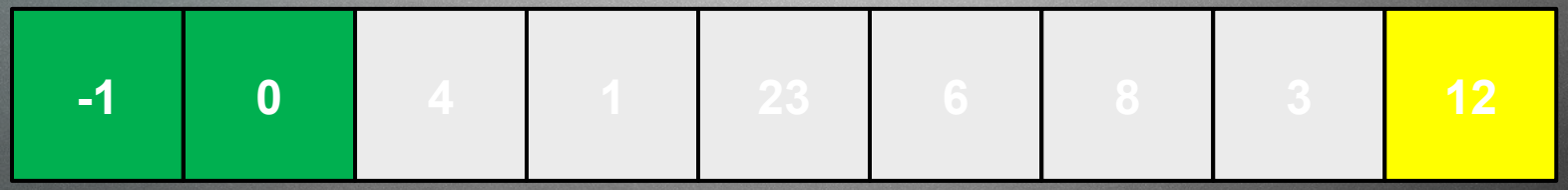
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

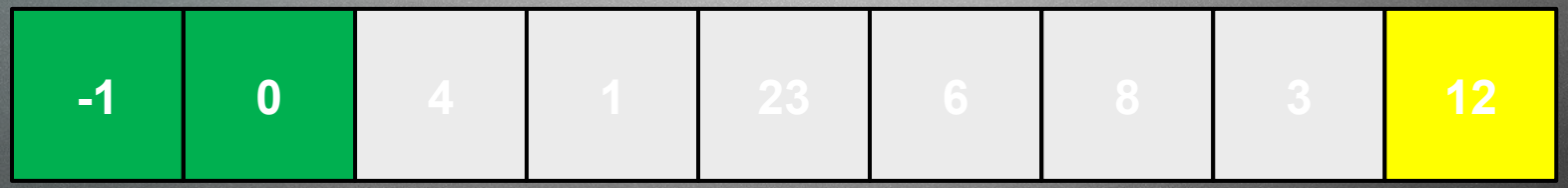
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

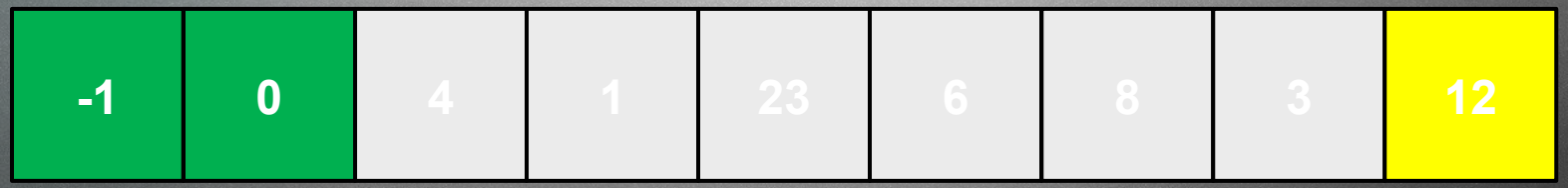
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=2

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

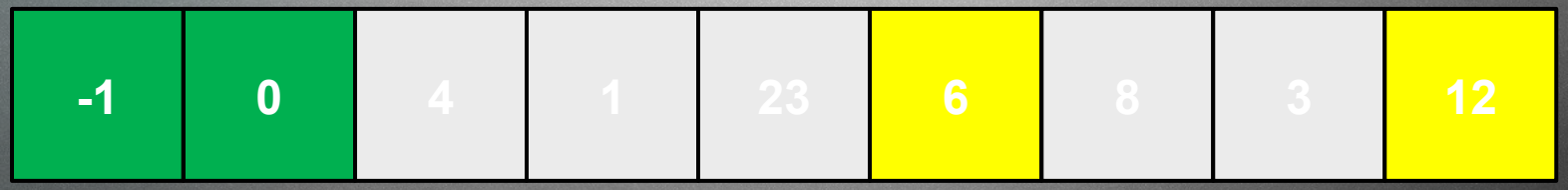
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=2

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

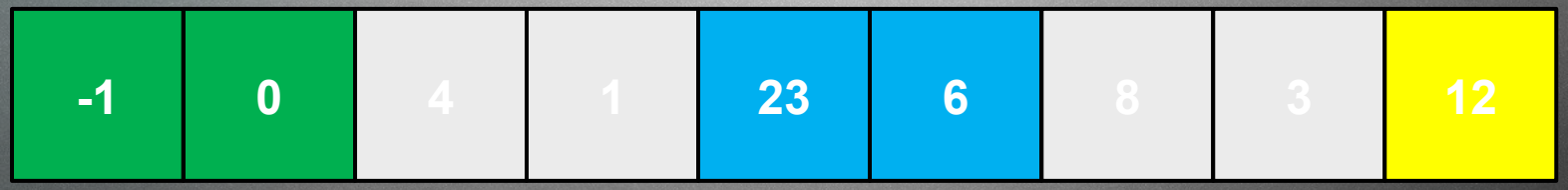
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=2

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

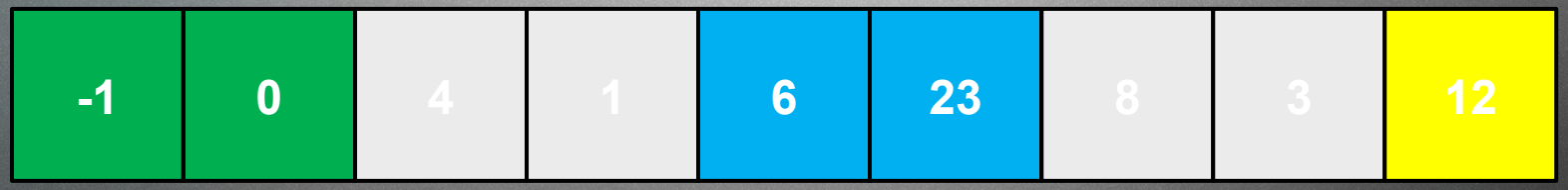
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=2

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

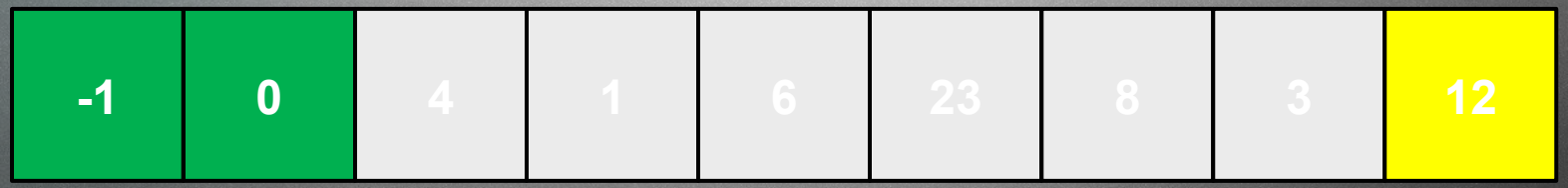
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=2

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

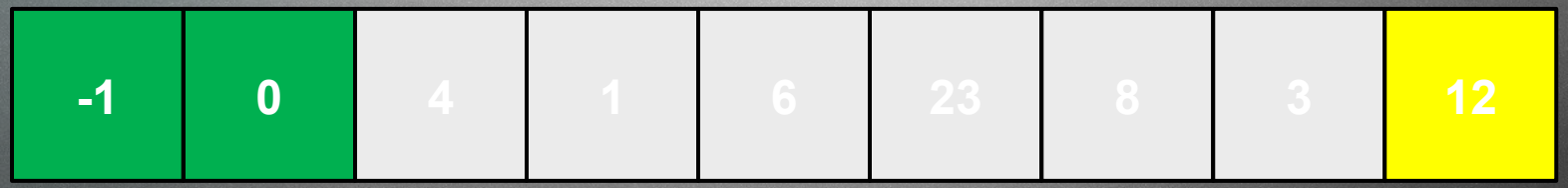
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=2

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=2

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	1	6	23	8	3	12

i=3

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

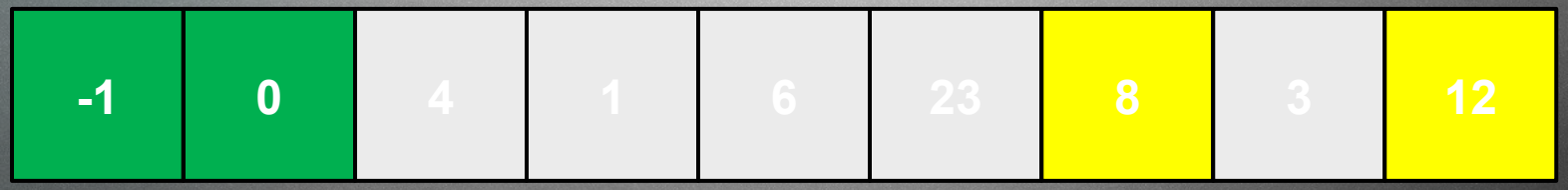
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	1	6	23	8	3	12

i=3

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

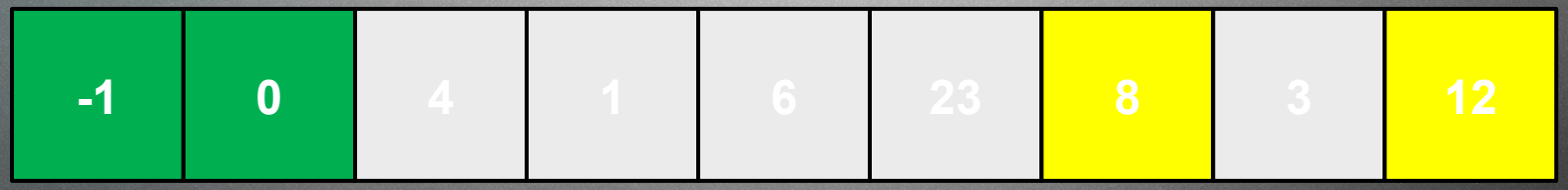
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=3

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

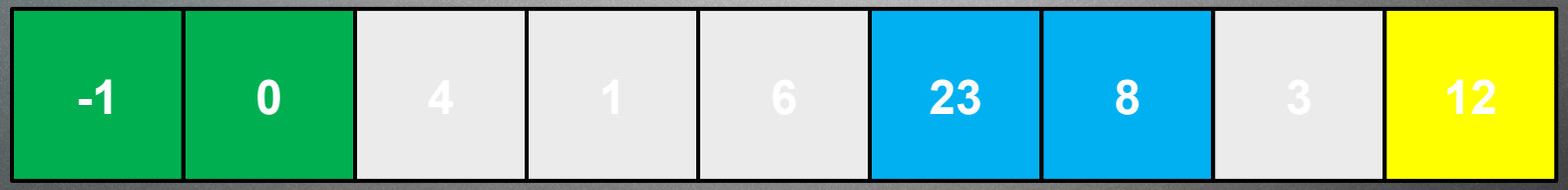
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=3

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

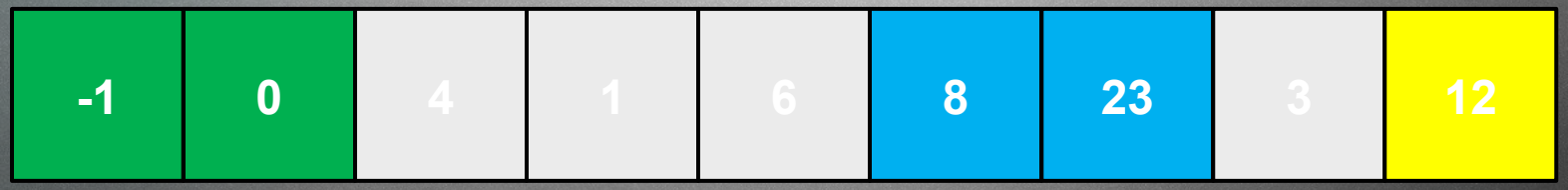
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=3

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=3

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	1	6	8	23	3	12

i=3

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	1	6	8	23	3	12

i=3

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	1	6	8	23	3	12

i=4

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	1	6	8	23	3	12

i=4

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	1	6	8	23	3	12

i=4

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

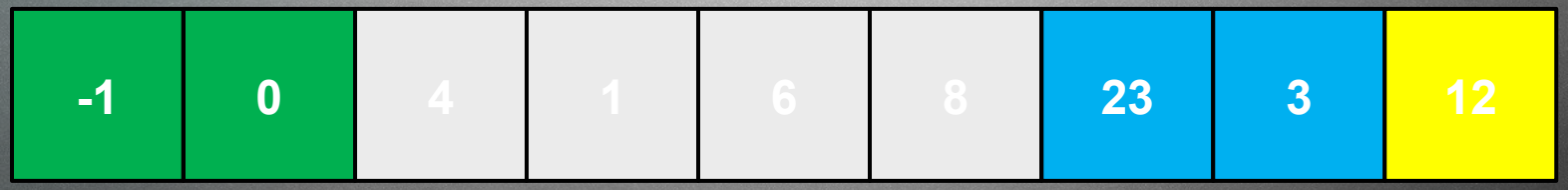
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	1	6	8	23	3	12

i=4

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

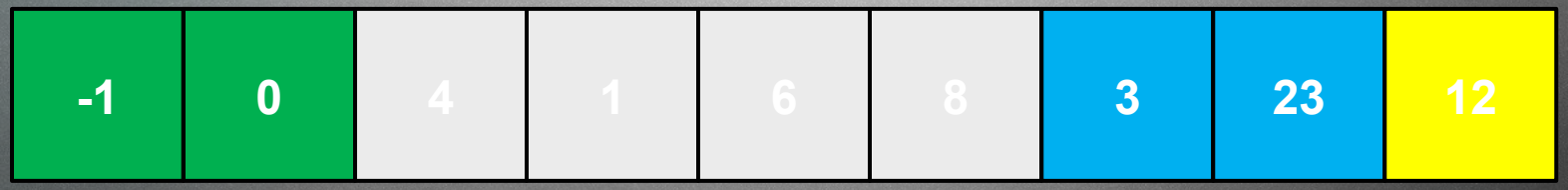
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=4

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=4

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	1	6	8	3	23	12

i=4

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

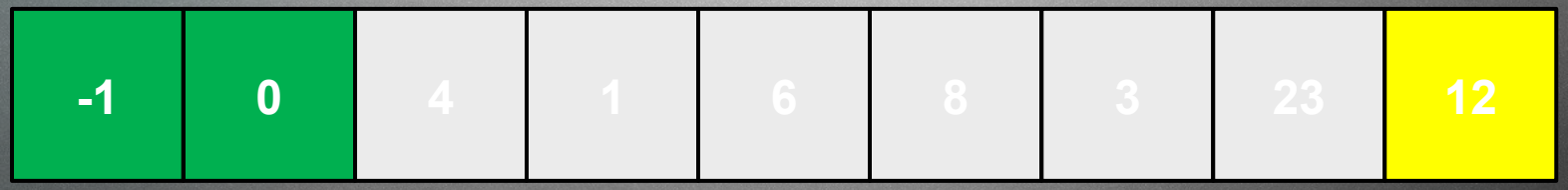
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	1	6	8	3	23	12

i=4

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

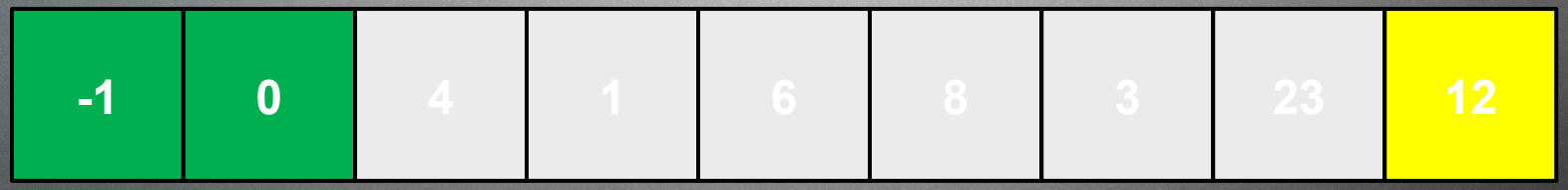
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=5

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

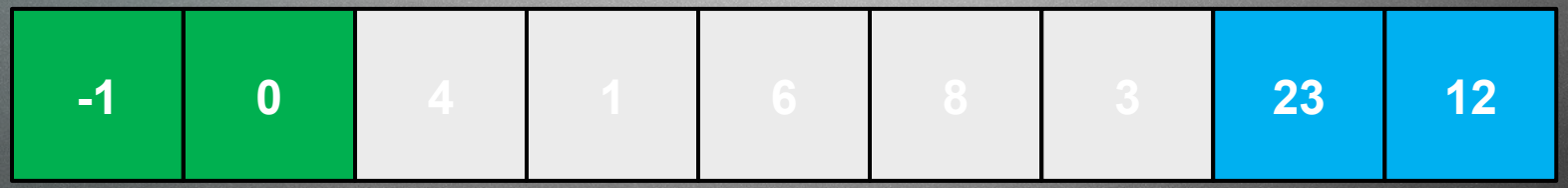
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=5

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

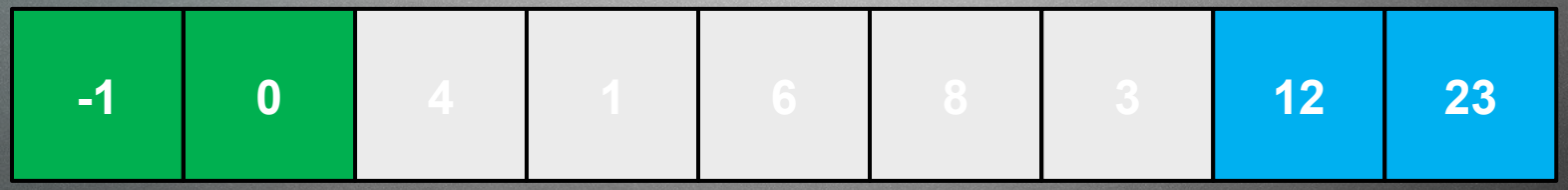
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=5

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=5

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

So: we have managed to rearrange the array so that on the left side of the pivot → items that are smaller and on the right are the greater items

-1	0	4	1	6	8	3	12	23

i=5

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

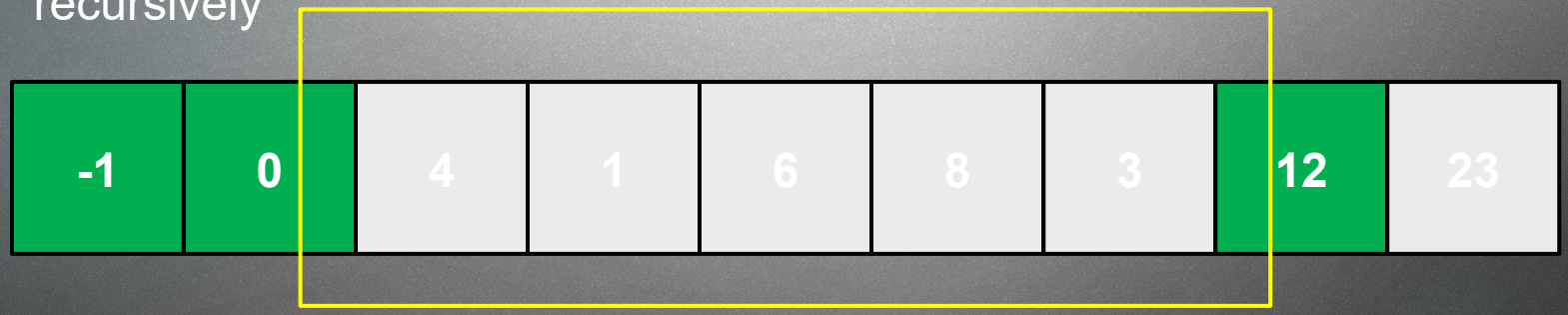
So: we have managed to rearrange the array so that on the left side of the pivot → items that are smaller and on the right are the greater items

-1	0	4	1	6	8	3	12	23

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

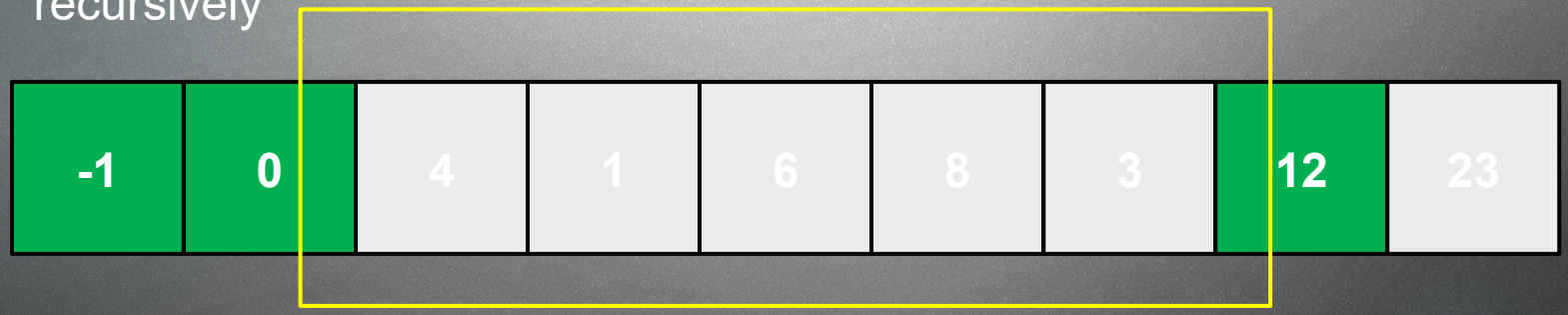
Sort left subarray recursively



```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

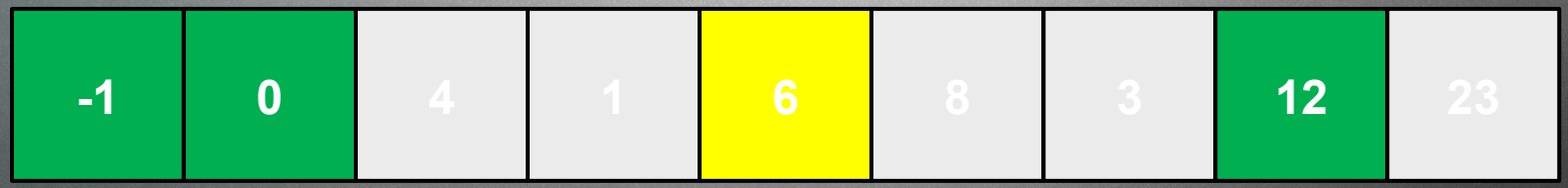
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

Sort left subarray recursively



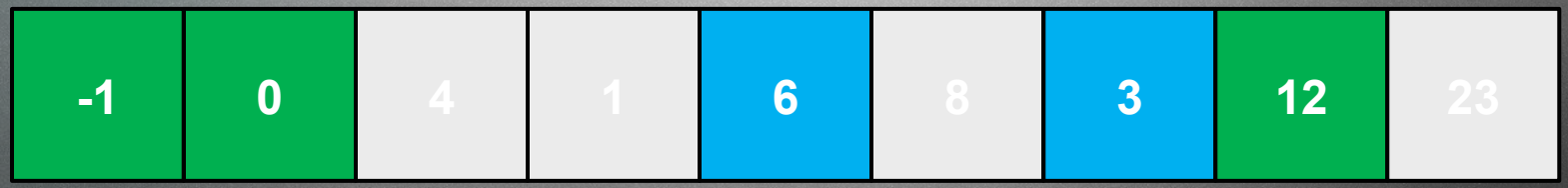
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



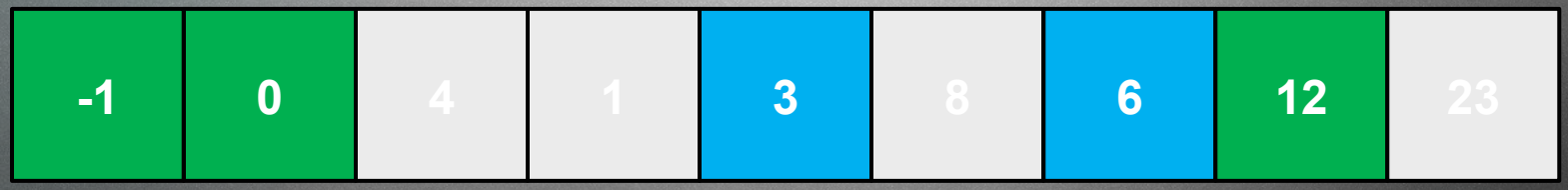
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	1	3	8	6	12	23

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	1	3	8	6	12	23

i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	1	3	8	6	12	23

i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

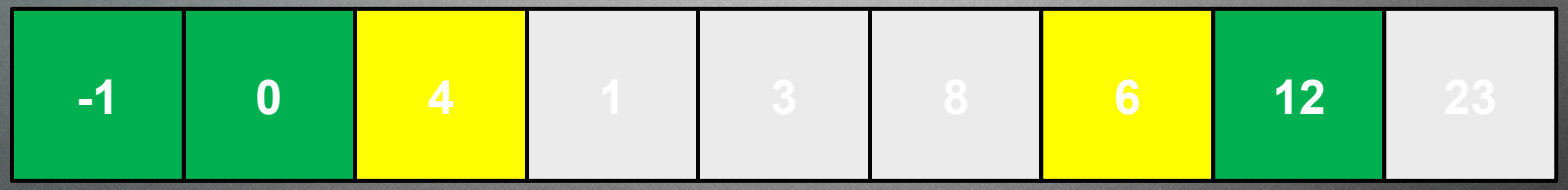
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	1	3	8	6	12	23

i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

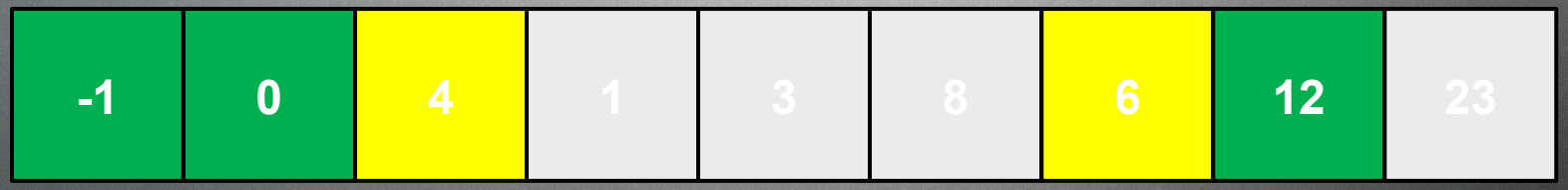
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

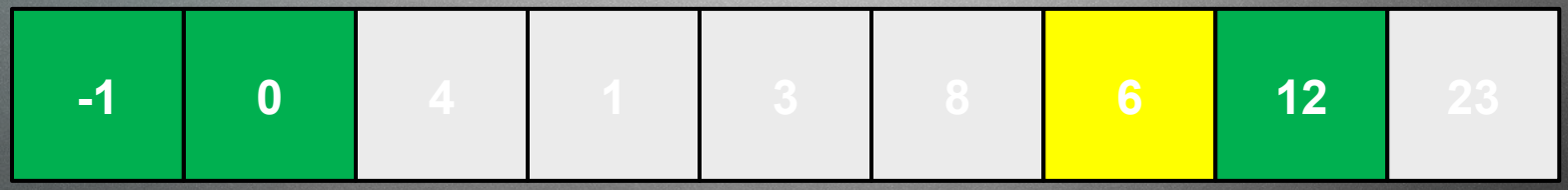
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

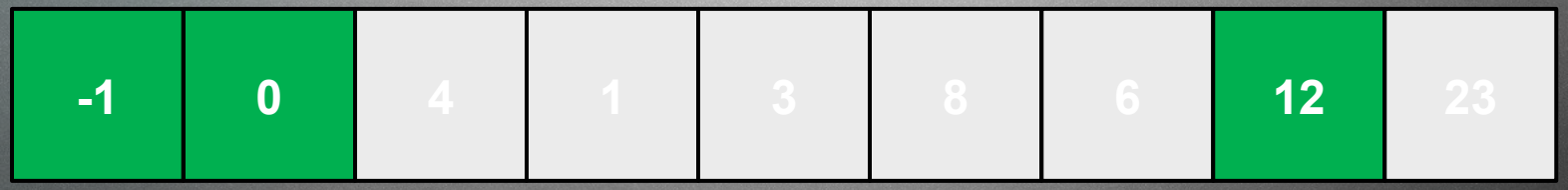
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

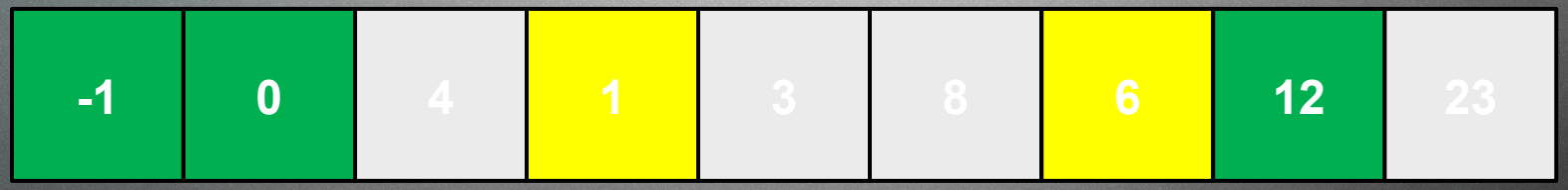
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

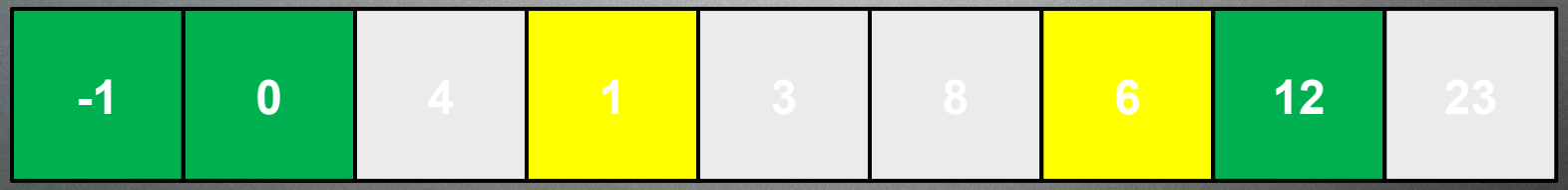
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

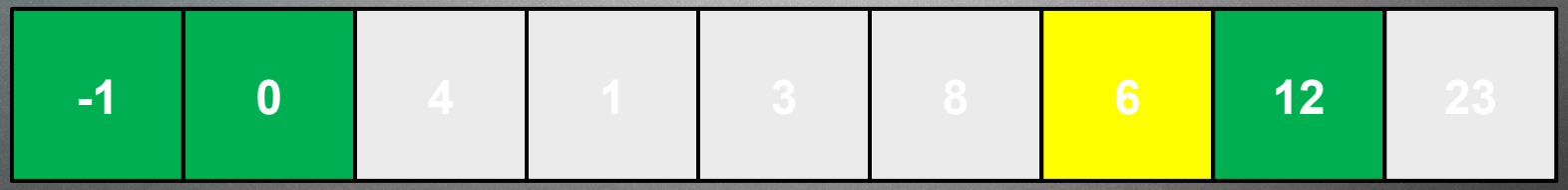
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=1

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

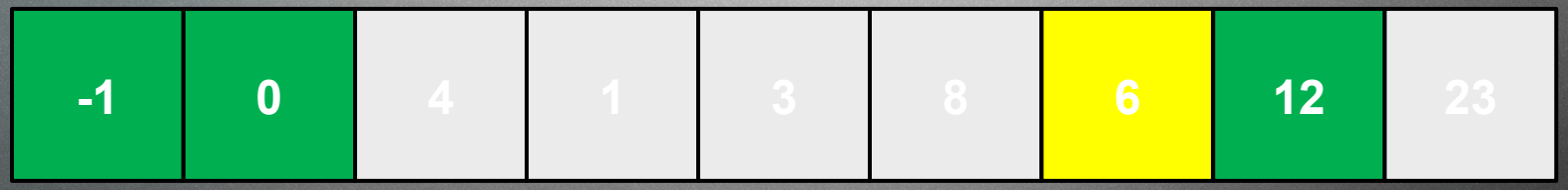
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=2

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

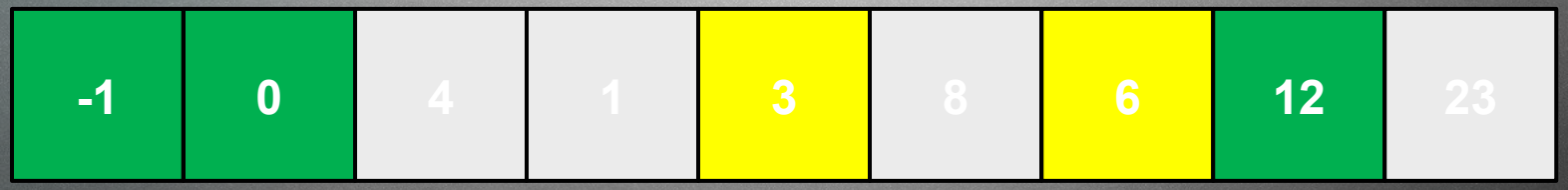
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=2

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

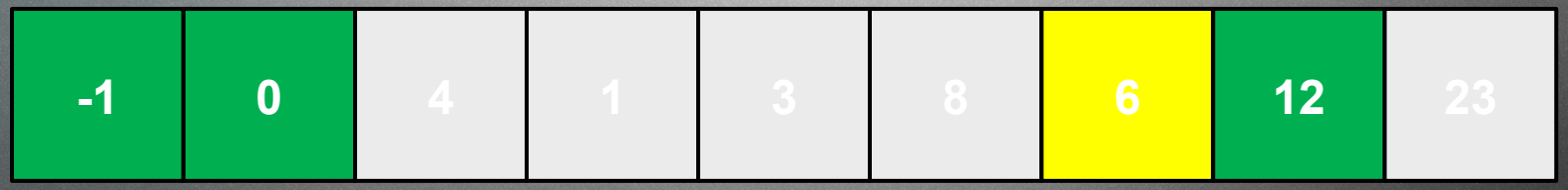
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=2

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

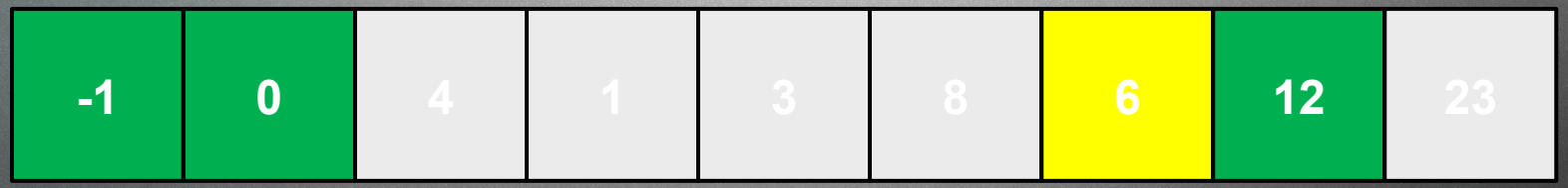
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=2

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

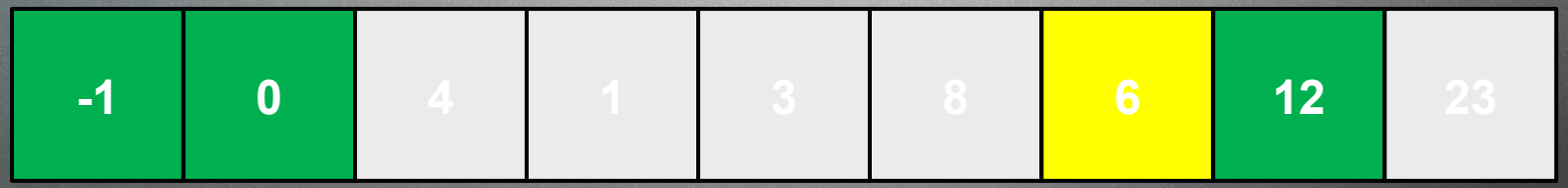
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=3

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

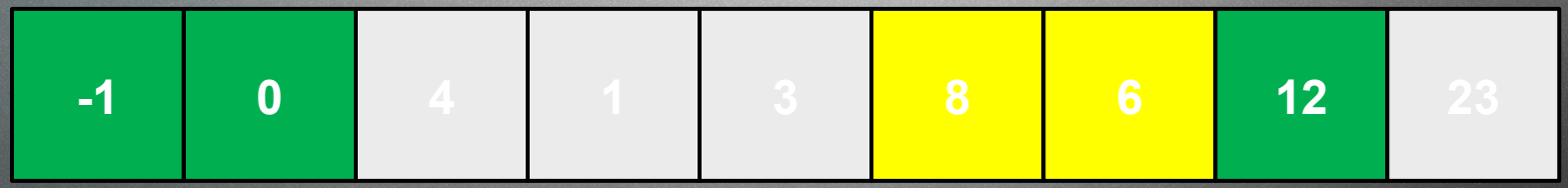
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=3

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=3

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	1	3	8	6	12	23

i=3

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

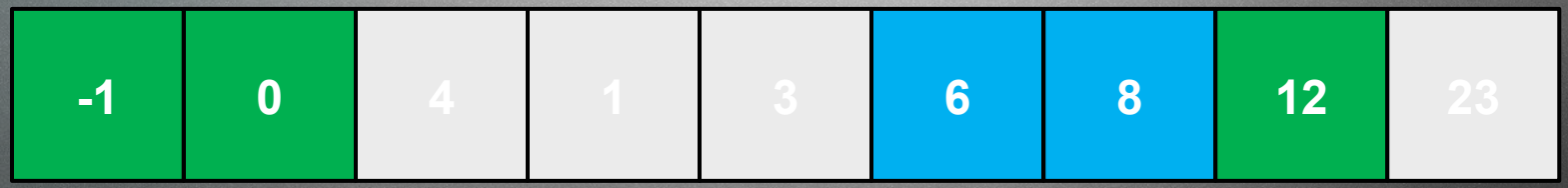
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	1	3	8	6	12	23

i=3

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=3

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

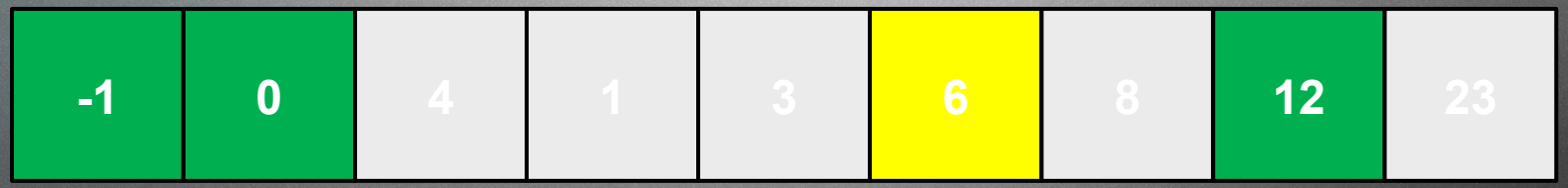
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	1	3	6	8	12	23

i=3

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=3

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

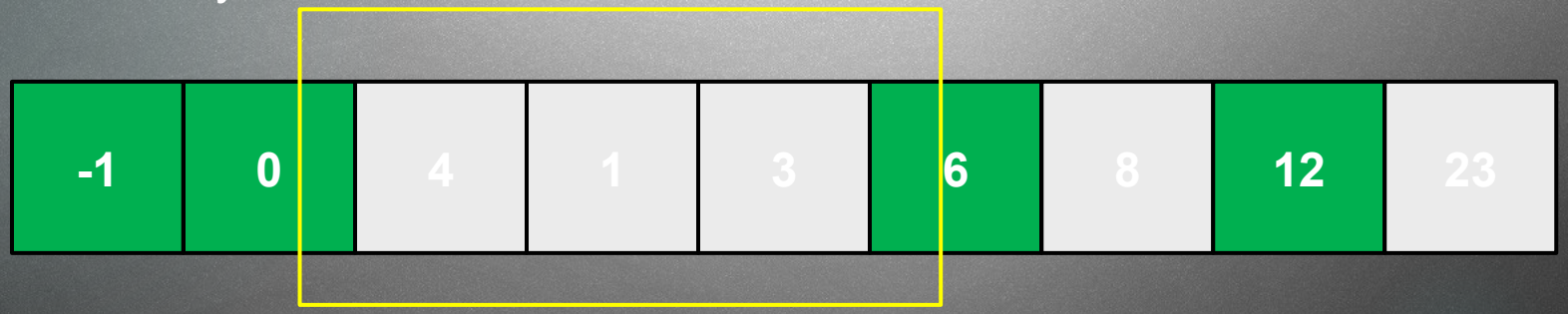
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	1	3	6	8	12	23

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

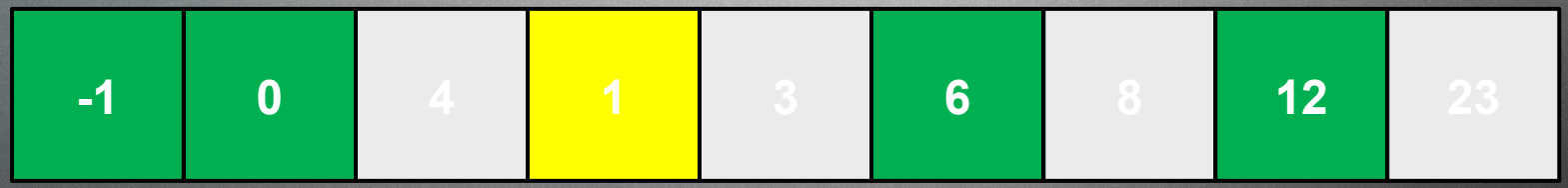
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

Sort left subarray recursively



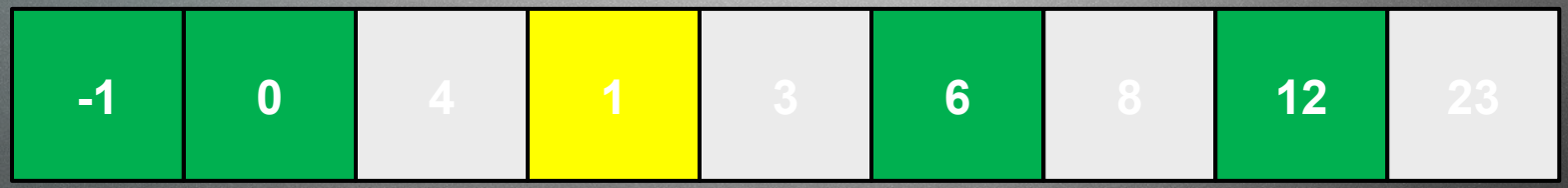
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

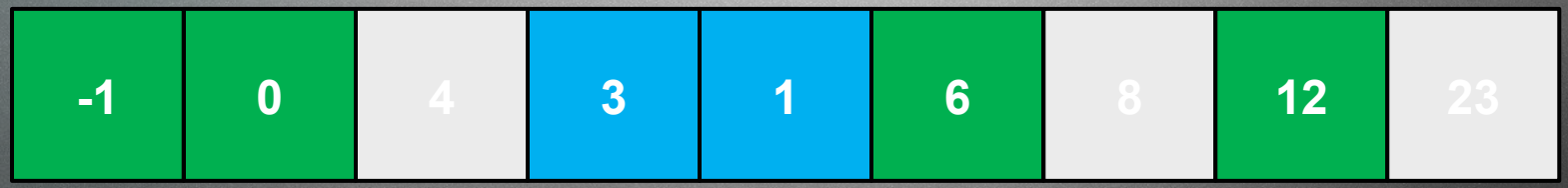


```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



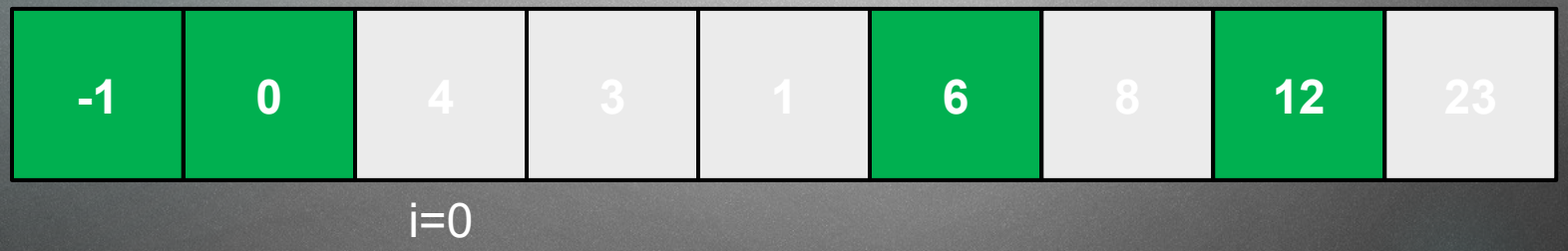
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	4	3	1	6	8	12	23

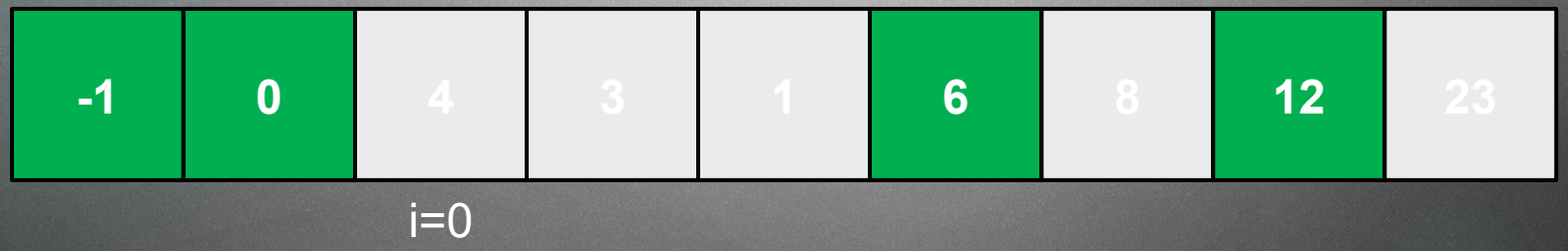
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



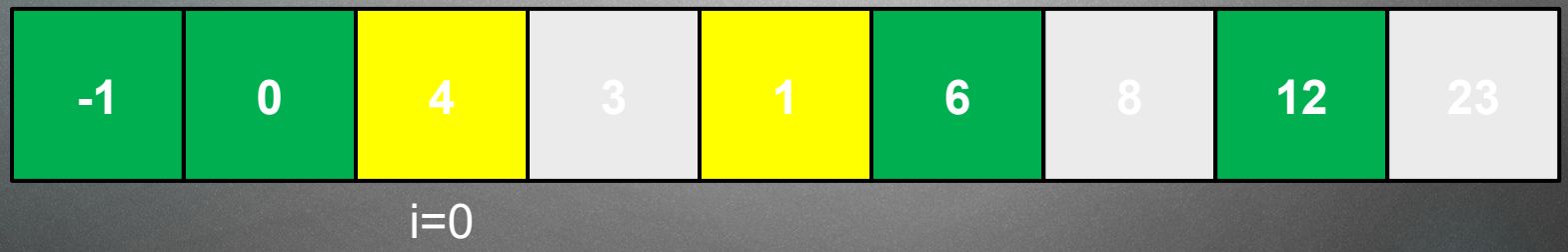
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



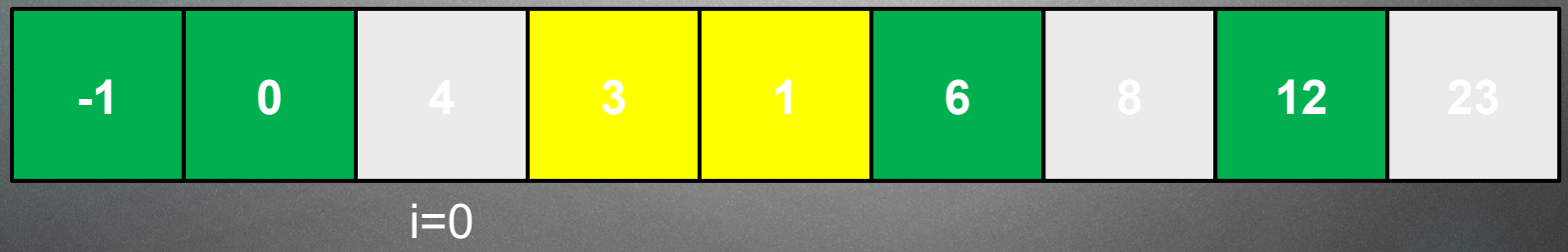
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



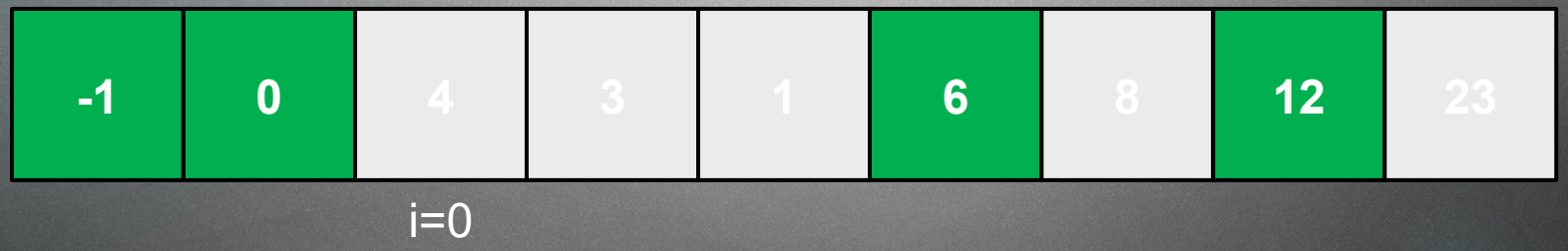
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



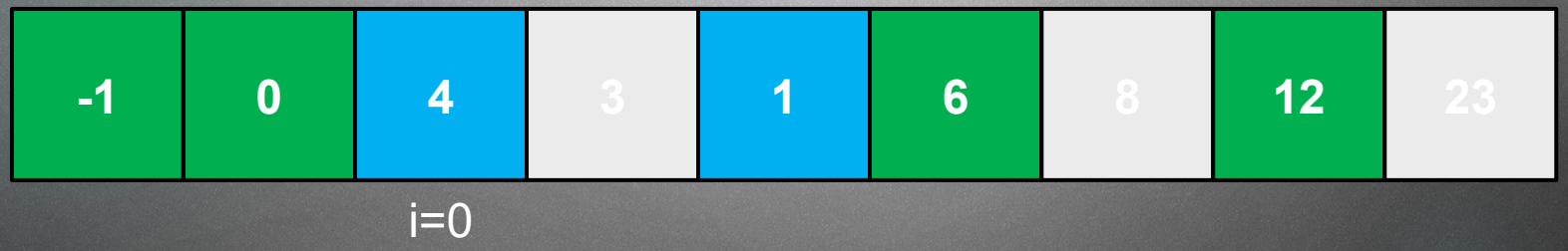
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



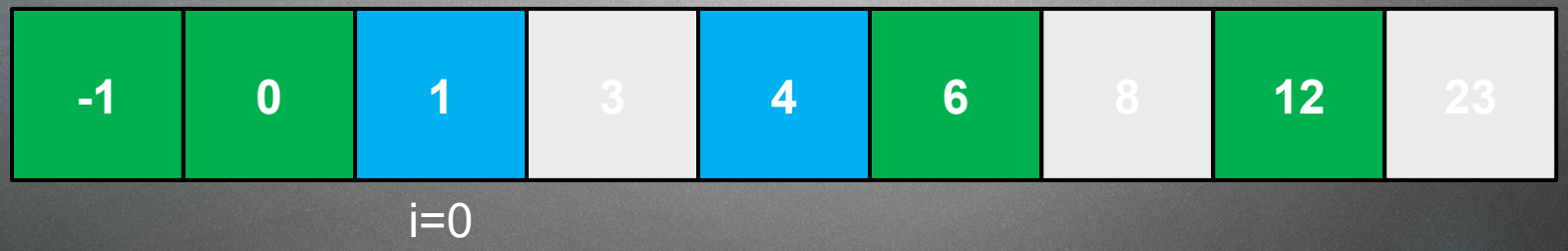
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	1	3	4	6	8	12	23

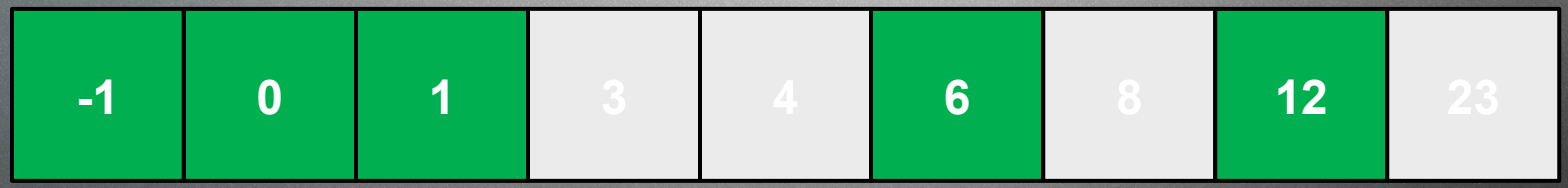
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	1	3	4	6	8	12	23

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

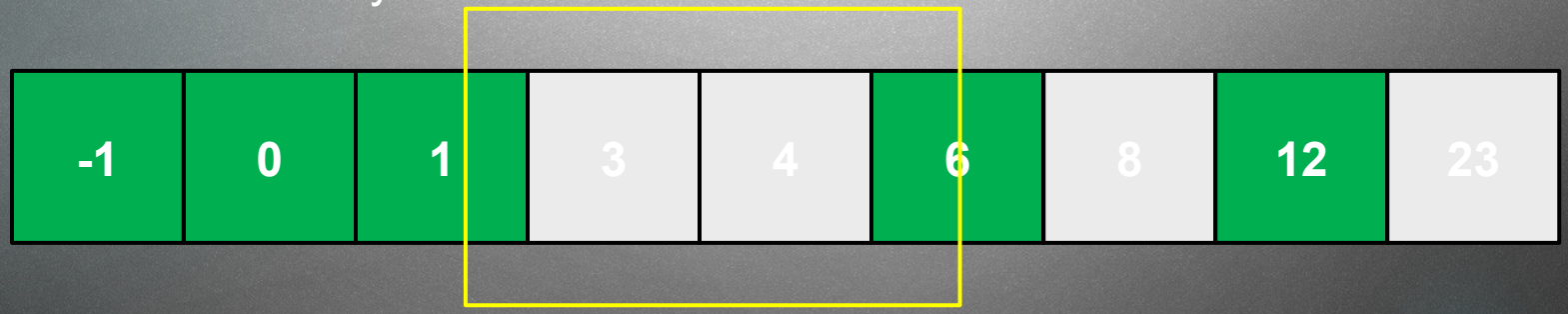
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

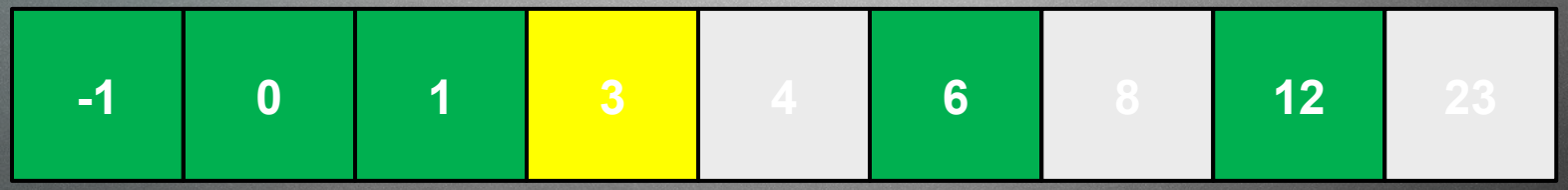
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

Sort right subarray recursively



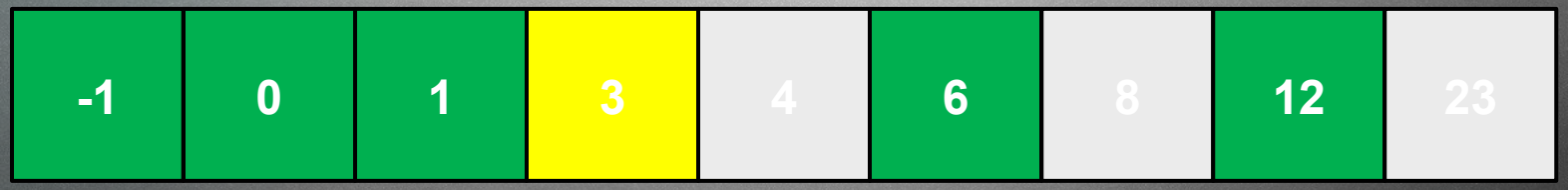
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

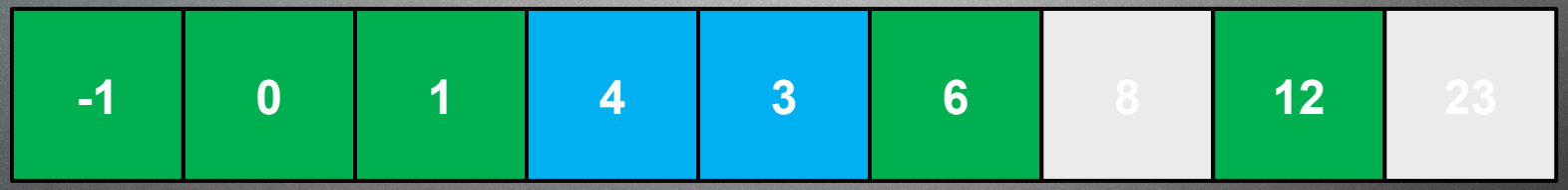


```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

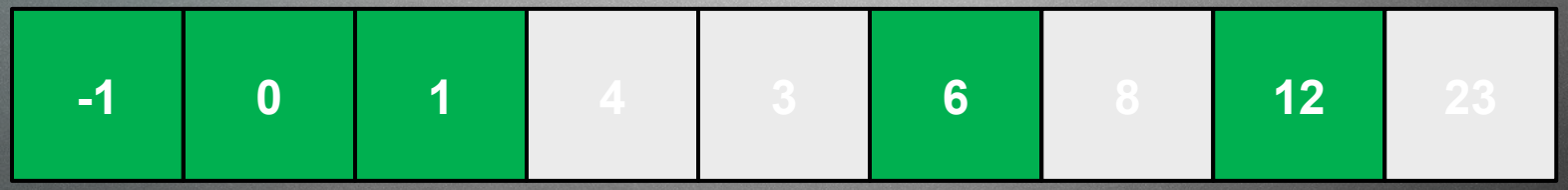


```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

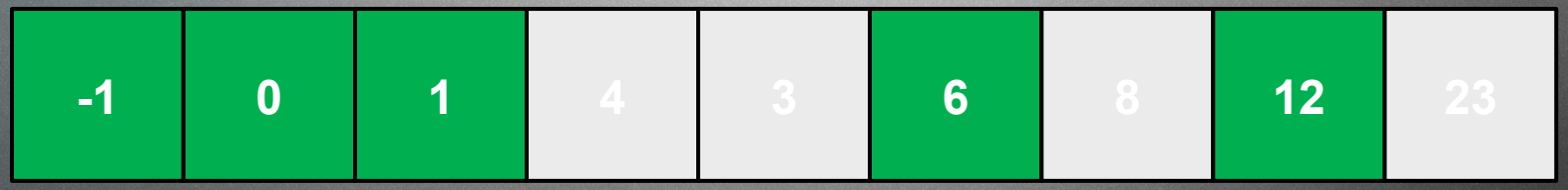
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

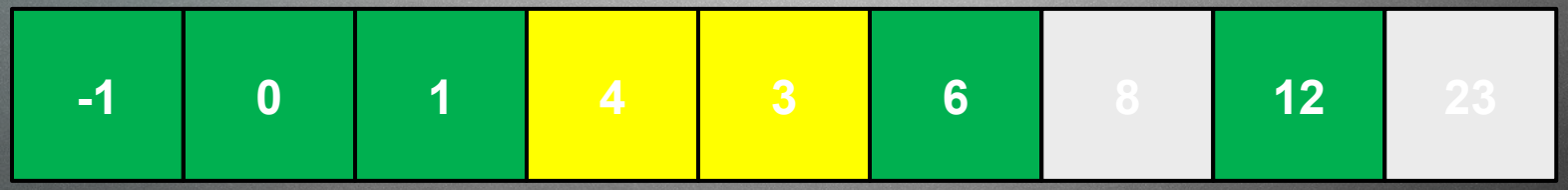
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

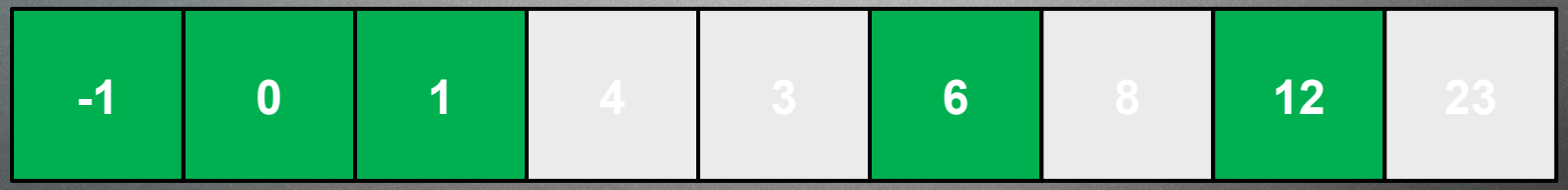
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

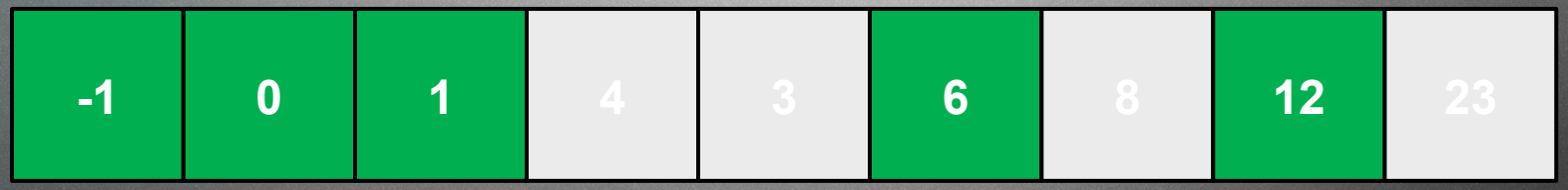
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

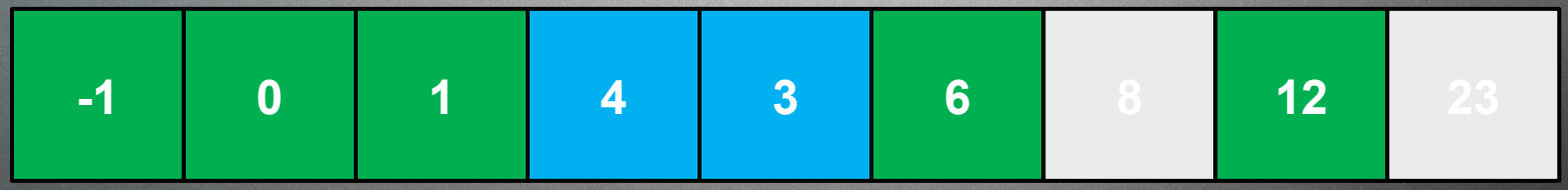
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=0

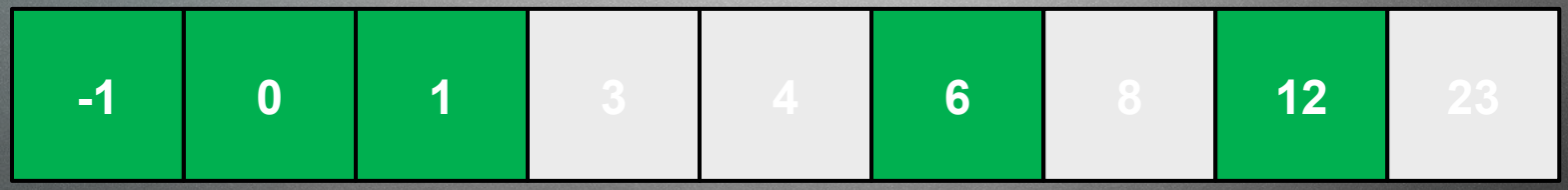
```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

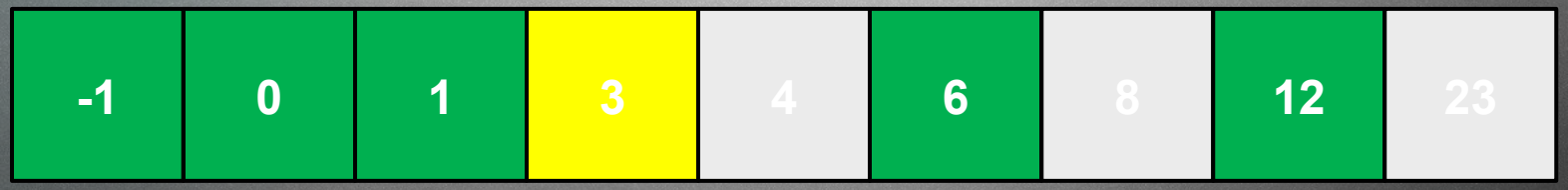
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

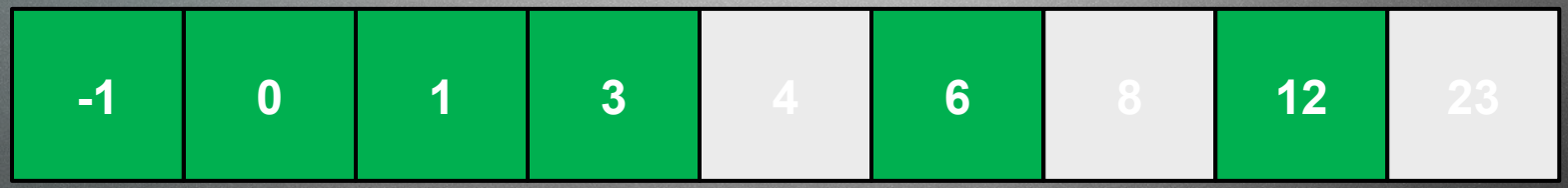
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



i=0

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

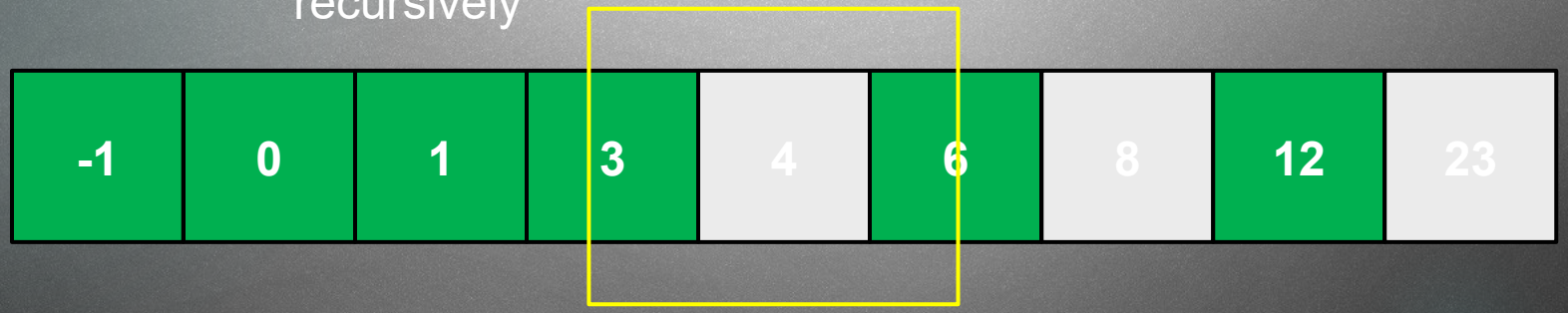
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

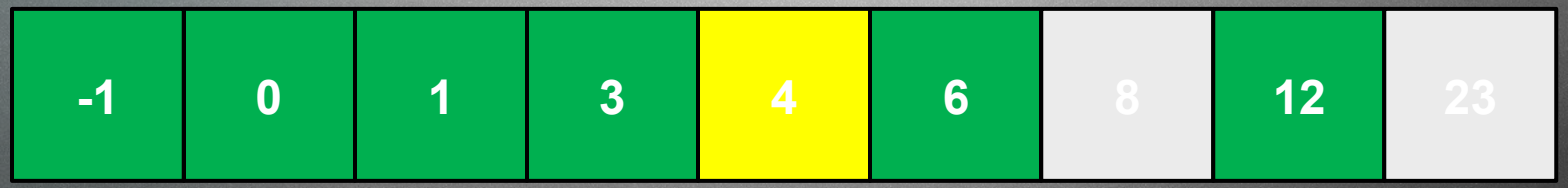
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

Sort right subarray recursively



```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

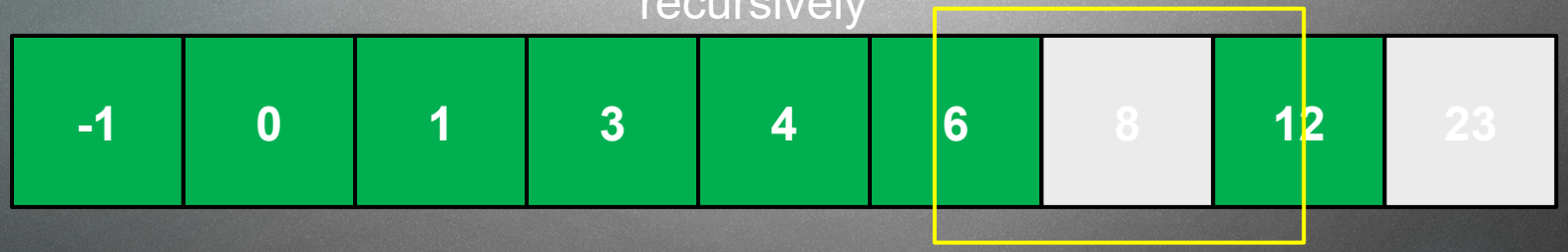
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	1	3	4	6	8	12	23

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

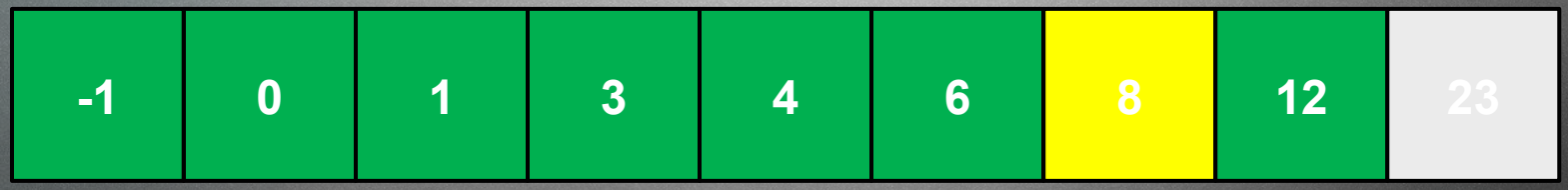
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

Sort right subarray recursively



```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```



```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

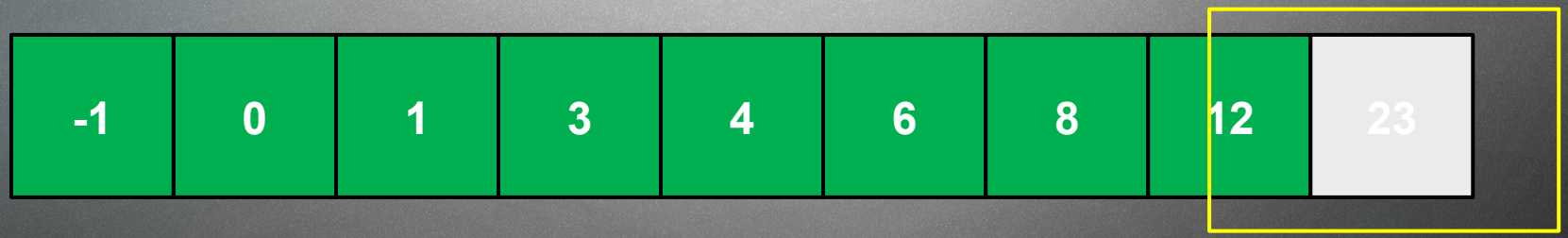
```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	1	3	4	6	8	12	23

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

Sort right subarray recursively



```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	1	3	4	6	8	12	23

```
quicksort(array,low,high)

    if low >=  high  return

    pivot = partition(array,low,high)
    quicksort(array,low,pivot-1)
    quicksort(array,pivot+1,high)

end
```

```
partition(array,low,high)

    pivotIndex = (low+high) / 2
    swap(pivotIndex,high)

    i = low

    for j=low to high
        if array[j] <= array[high]
            swap(i,j)
            i++

    swap (i,high)

    return i
end
```

-1	0	1	3	4	6	8	12	23

Worst case

The most naive pivot selection algorithm is to choose the first element as the pivot

~ this results in worst case behavior if the data is already sorted

(the first element will always be the min)

$$O(N^2)$$