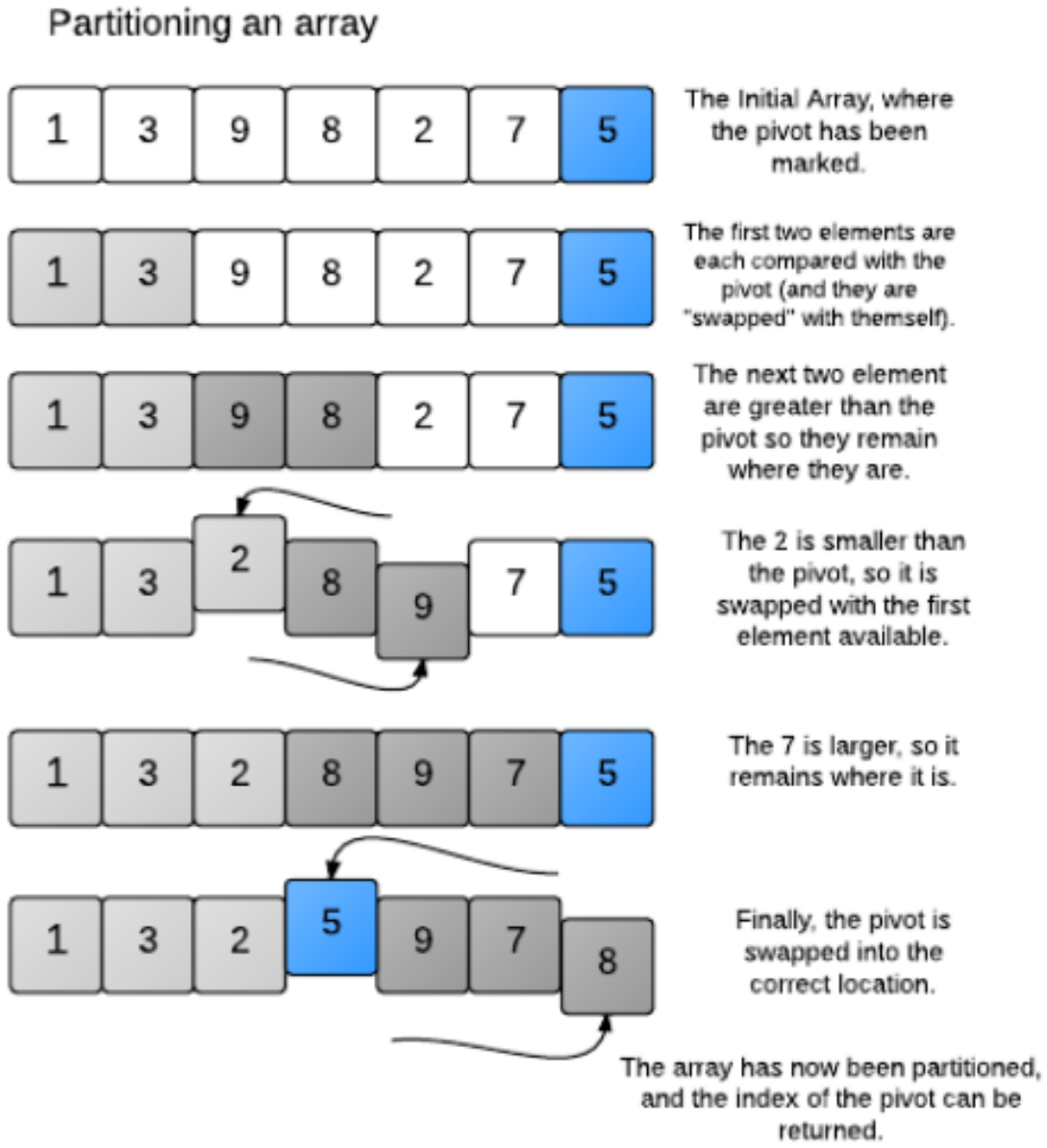
Quick Sort

Labwork 9

Quick Sort


Partitioning an array
1 3 9 8 2 7 5
The Initial Array, where the pivot has been marked.
1 3 9 8 2 7 5
The first two elements are each compared with the pivot (and they are "swapped" with themself).
1 3 9 8 2 7 5
The next two element are greater than the pivot so they remain where they are.
1 3 2 8 9 7 5
The 2 is smaller than the pivot, so it is swapped with the first element available.
1 3 2 8 9 7 5
The 7 is larger, so it remains where it is.
1 3 2 5 9 7 8
Finally, the pivot is swapped into the correct location.
The array has now been partitioned, and the index of the pivot can be returned.

Pseudocode

```
Sort(A)
        Quicksort(A,1,n)


Quicksort(A, low, high)
        if (low < high)
                pivot-location = Partition(A,low,high)
                Quicksort(A,low, pivot-location - 1)
                Quicksort(A, pivot-location+1, high)


Partition(A,low,high)
        pivot = A[low]
        leftwall = low
        for i = low+1 to high
                if (A[i] < pivot) then
                        leftwall = leftwall+1
                        swap(A[i],A[leftwall])
        swap(A[low],A[leftwall])
```