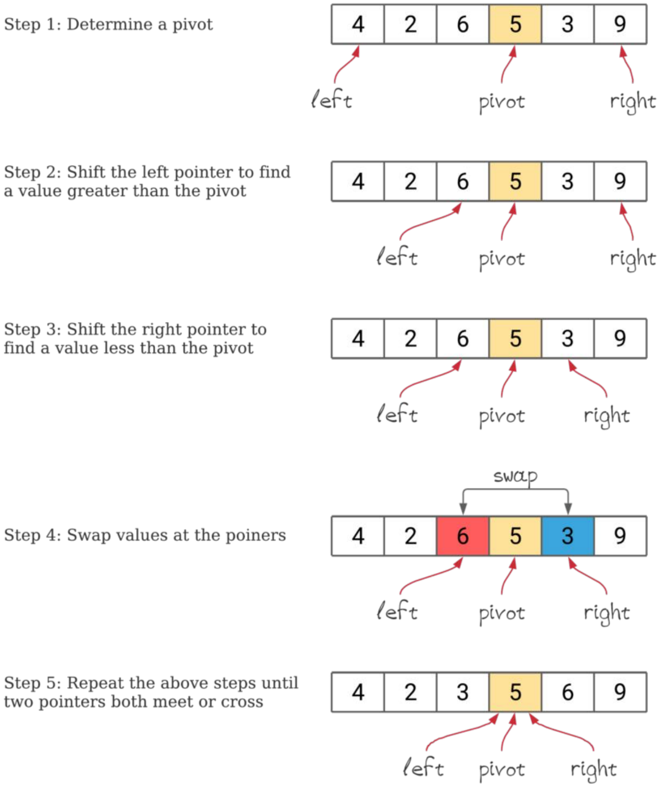
Quicksort (group 2)

Max Ryoo, Maggie Houck, Kaia Lindberg, and Cepehr Alizadeh

General description of the algorithm

- Based on divide and conquer paradigm
- First, partitions the array into two parts
- Then, sorts the two parts independently
- Finally, it combines the two sorted parts via a simple concatenation

Demonstration: How it works (review code/algorithm)



Step 1: Determine a pivot

4	2	6	5	3	9

Step 2: Shift the left pointer to find a value greater than the pivot

4	2	6	5	3	9

Step 3: Shift the right pointer to find a value less than the pivot

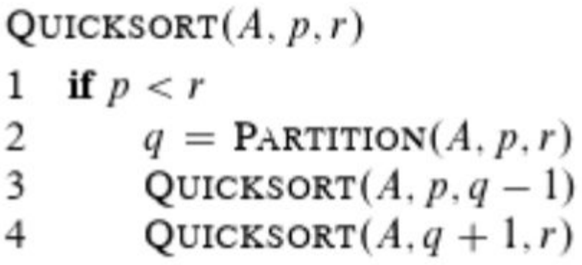
4	2	6	5	3	9

Step 4: Swap values at the poiners

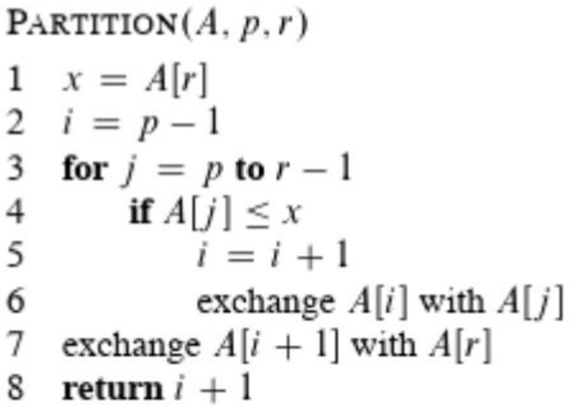
4	2	6	5	3	9

Step 5: Repeat the above steps until two pointers both meet or cross

4	2	3	5	6	9



```
QUICKSORT$(A, p, r)$
if $p < r$
    $q$ = PARTITION$(A, p, r)$
    QUICKSORT$(A, p, q - 1)$
    QUICKSORT$(A, q + 1, r)$
```



```
PARTITION$(A, p, r)$
$x = A[r]$
$i = p - 1$
for $j = p$ to $r - 1$
    if $A[j] \leq x$
        $i = i + 1$
        exchange $A[i]$ with $A[j]$
exchange $A[i + 1]$ with $A[r]$
return $i + 1$
```

Advantages?

- Quicksort is an inplace sorting algorithm so no new space complexity is created.
- In practice, quicksort is often laster than other O(nlogn) algorithms

Disadvantages?

- Quicksort has a worst complexity of O(n^2). In some cases we are losing complexity time when compared to a better worst case complexity sorting algorithm such as mergeSort
- It has to utilize recursion
- Not always stable

Complexity (in the worst case—that is Big-Oh!)

- Worst case complexity is n^2
  - This can happen when every item of the array is already in order
  - When the pivot is an extreme, requiring the maximum number of iterations
- Best and average case is usually n log (n)
  - Log part comes from dividing into subsequences

Would you recommend this algorithm? Why or why not?

- Primarily recommend its uses for information sorting due to its speed
- Also ideal for commercial computation settings (ex. organizations sorting accounts by name or ID)
- For shorter array sizes quicksort is more efficient than say Merge Sort. If the problem is for smaller cases and “quick” sorting then quicksort can be a solution.
- For situations where the users determine the order to sort by, quicksort may not be the most ideal solution

References

https://www.tutorialspoint.com/design_and_analysis_of_algorithms/design_and_analysis_of_algorithms_quick_sort.htm

https://www.geeksforgeeks.org/application-and-uses-of-quicksort/#:~:text=The%20sorting%20algorithm%20is%20used,reference%20when%20used%20for%20arrays.

https://www.crio.do/blog/top-10-sorting-algorithms/

https://www.tutorialspoint.com/design_and_analysis_of_algorithms/design_and_analysis_of_algorithms_quick_sort.htm

https://www.interviewbit.com/tutorial/quicksort-algorithm/