
Algorithm 13: QuickSort(A)

Input: array $A = (a_0, \dots, a_{n-1})$

Output: sorted array A

```
1 if length( $A$ )  $\leq$  1 then
2   | return  $A$ 
3  $p \leftarrow$  pivot element from  $A$ 
4 remove  $p$  from  $A$ 
5  $L \leftarrow \emptyset, G \leftarrow \emptyset$ 
6 for  $i \leftarrow 1, \dots, n - 1$  do
7   | if  $a_i \leq p$  then
8     | add  $a_i$  to  $L$ 
9   | else
10  | add  $a_i$  to  $G$ 
11 return concat(QuickSort( $L$ ),  $p$ , QuickSort( $G$ ))
```

Fun Fact to Know: QuickSort in Haskell:

```
quicksort :: Ord a => [a] -> [a]
quicksort (p:A) = (quicksort [x | x <- A, x < p])
                  ++ [p]
                  ++ (quicksort [x | x <- A, x >= p])
quicksort [] = []
```

This is working code!