Quicksort

Eingabe: Array A mit n Elementen

Ausgabe: sortiertes Array A

QUICKSORT(A)

(1) verschiebe die Elemente in A so, dass sich im vorderen Teil des Arrays „kleine" und im hinteren Teil „große" Elemente befinden;

(2) sortiere den vorderen und den hinteren Teil rekursiv;

(3) hänge beide Teile aneinander;

Eingabe: Array A mit n Elementen; Index p des Pivotelements, $1 \leq p \leq n$

Ausgabe: Index q des Pivotelementes in A, wobei gilt $A_k < A_q$ für $1 \leq k < q$ und $A_q < A_l$ für $q < l \leq n$

PARTITION(A, p)

/* ... */

Eingabe: Array A mit n Elementen

Ausgabe: sortiertes Array A

QUICKSORT(A)

    $q :=$ PARTITION(A, n);
    if $q - 1 \leq n - q$ **then**
        QUICKSORT($(A_1, \ldots, A_{q-1})$);
        QUICKSORT($(A_{q+1}, \ldots, A_n)$);
    else
        QUICKSORT($(A_{q+1}, \ldots, A_n)$);
        QUICKSORT($(A_1, \ldots, A_{q-1})$);
    end
    return A;

Eingabe: Array A mit n Elementen; Index p des Pivotelements, $1 \leq p \leq n$

Ausgabe: Index q des Pivotelementes in A, wobei gilt $A_k < A_q$ für $1 \leq k < q$ und $A_q < A_l$ für $q < l \leq n$

PARTITION(A, p)

$i := 0$; $j := n$;
SWAP(A_p, A_n); pivot $:= A_n$;
while $i < j$ **do**
    repeat $i++$ **until** $i \geq j$ **or** $A_i >$ pivot;
    repeat $j--$ **until** $i \geq j$ **or** $A_j <$ pivot;
    if $i < j$ **then** SWAP(A_i, A_j);
end
SWAP(A_i, A_n);
return i;