

Algoritmo de ordenação QuickSort, por Rian Beskow Friedrich:

```
def quicksort(arr):  
    if len(arr) <= 1:  
        return arr  
    else:  
        pivot = arr[len(arr) // 2]  
        left = [x for x in arr if x < pivot]  
        middle = [x for x in arr if x == pivot]  
        right = [x for x in arr if x > pivot]  
        return quicksort(left) + middle + quicksort(right)  
  
lista = [10, 7, 8, 9, 1, 5]  
print("Lista original:", lista)  
print("Lista ordenada:", quicksort(lista))
```

Contagem de instruções:

Caso Base: Quando o algoritmo não precisa ordenar nada, apenas retorna o array.

Número de instruções: 3.

Caso Geral: Quando o array tem tamanho maior que 1 ($\text{len}(\text{arr}) > 1$) e o algoritmo escolhe um pivot, dividindo a lista em três sublistas e chama recursivamente o quicksort nas sublistas.

Número de instruções fora das chamadas recursivas: aproximadamente $7n + 8$.

Sendo n o tamanho do array ou lista que está sendo processado.

Polinômio:

$$P(n) = 4n + 3$$