

Quick Sort 

Quick Sort

1. Wybierz pivot p
2. Podziel tablicę T względem pivotu P umieszczając mniejsze wartości od P po lewej a większe - po prawej. Nazywamy tą funkcję *partition*.



3. Rekurencyjnie wywołaj *quicksort* na lewej części tablicy
4. Rekurencyjnie wywołaj *quicksort* na prawej części tablicy

Hoare's Partition

Jest to najpopularniejsza oraz najszybsza implementacja funkcji *partition*, która pozwala na to, że Quick Sort ma złożoność $O(n \log n)$.

Główne założenie ogólnej funkcji *partition* jest to, żeby pivot P przesunąć na jego miejsce w posortowanej tablicy, a więc musi mieć same elementy mniejsze po lewej stronie jak i tylko elementy większe po prawej stronie.

Algorytm.

1. Dla fragmentu tablicy na której operujemy $A[l \dots r]$ stwórzmy indeksy l i r (początek i koniec tablicy)

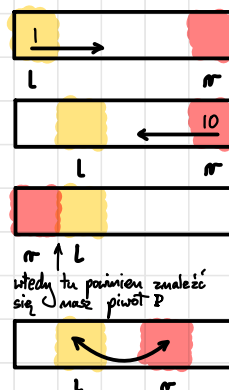
2. **while** $l < r$

a) **while** $A[l] < P : l++$

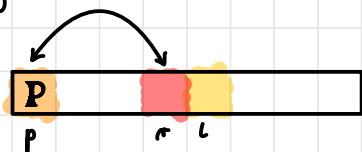
b) **while** $A[r] > P : r--$

c) **if** $r < l : \text{return } r$
przypadek gdy skończyliśmy

d) **swap** ($A[l], A[r]$)



W podpunkcie 2.c zwracamy miejsce nowego pivota.
W przypadku, gdy pivot przechowywujemy na początku tablicy możemy go zamienić z $A[r]$



Możemy tak zrobić, ponieważ w $A[r]$ mamy wartość mniejszą od pivota.

Charakterystyka

Hoare's Partition jest niestabilny. Co więcej sortowanie QuickSort jest niestabilne.