

QuickSort z Partition w wersji Lomuto z losowaniem pivota

```
def partitionright(arr, left, right):
    pivot = arr[right]
    j=left

    for i in range(left,right):
        if arr[i] < pivot:
            arr[j],arr[i]=arr[i],arr[j]
            j+=1
    arr[j],arr[right]=arr[right],arr[j]
    return j

def quicksort(arr, left, right):
    if left <= right:
        j=partitionright(arr, left, right)
        quicksort(arr, j+1, right)
        quicksort(arr, left, j-1)
    return arr
```

QuickSort z Partition w wersji Lomuto z losowaniem pivota jako mediany z trzech wartości z tablicy

```
def partitionmedian(arr, left, right):
    mid = (left + right)//2
    if arr[mid] < arr[left]:
        arr[left], arr[mid] = arr[mid], arr[left]
    if arr[right] < arr[left]:
        arr[left], arr[right] = arr[right], arr[left]
    if arr[mid] < arr[right]:
        arr[mid], arr[right] = arr[right], arr[mid]
    pivot = arr[right]
    j=left

    for i in range(left,right):
        if arr[i] < pivot:
            arr[j],arr[i]=arr[i],arr[j]
            j+=1
    arr[j],arr[right]=arr[right],arr[j]
    return j

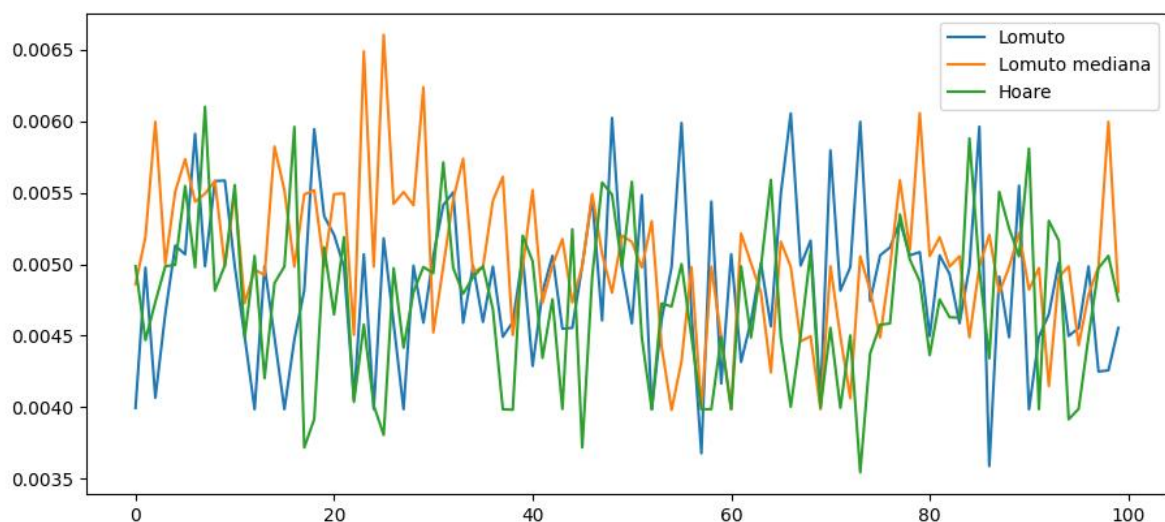
def quicksort2(arr, left, right):
    if left <= right:
        j=partitionmedian(arr, left, right)
        quicksort2(arr, j+1, right)
        quicksort2(arr, left, j-1)
    return arr
```

QuickSort z Partition w wersji Hoare (z dwoma indeksami idącymi naprzeciw siebie)

```
def partitionhoare(arr, left, right):
    pivot = arr[right]
    i = left
    j = right
    while i < j:
        while arr[i] < pivot:
            i+=1
        while arr[j] > pivot:
            j-=1
        if i < j:
            arr[i], arr[j] = arr[j], arr[i]
            i+=1
    return j

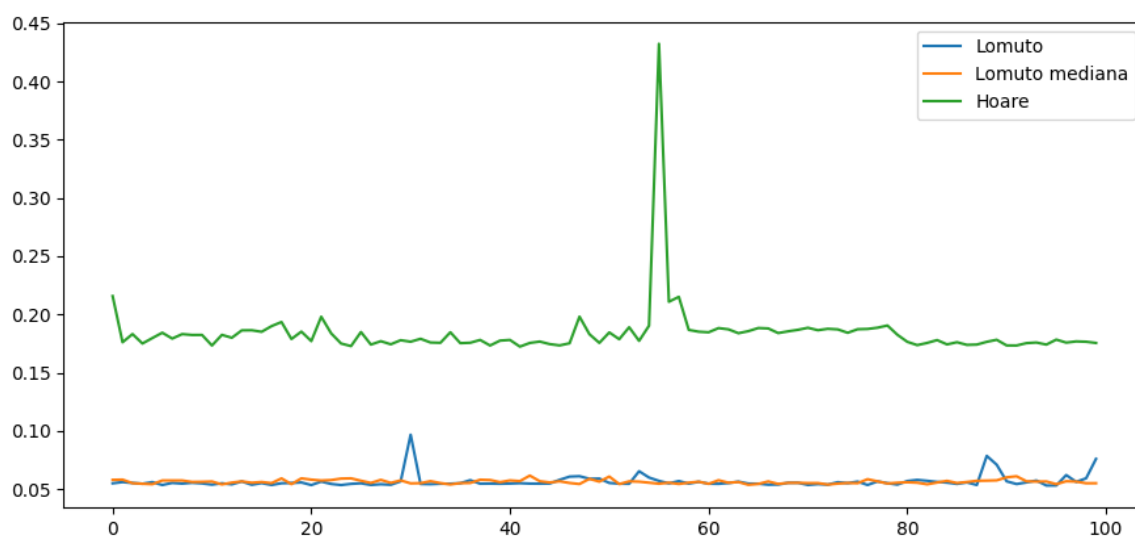
def quicksort3(arr, left, right):
    if left <= right:
        j=partitionhoare(arr, left, right)
        quicksort3(arr, j + 1, right)
        quicksort3(arr, left, j - 1)
    return arr
```

Porównanie działania na losowych listach 5000 elementowych zawierających liczby od 0 do 1000.



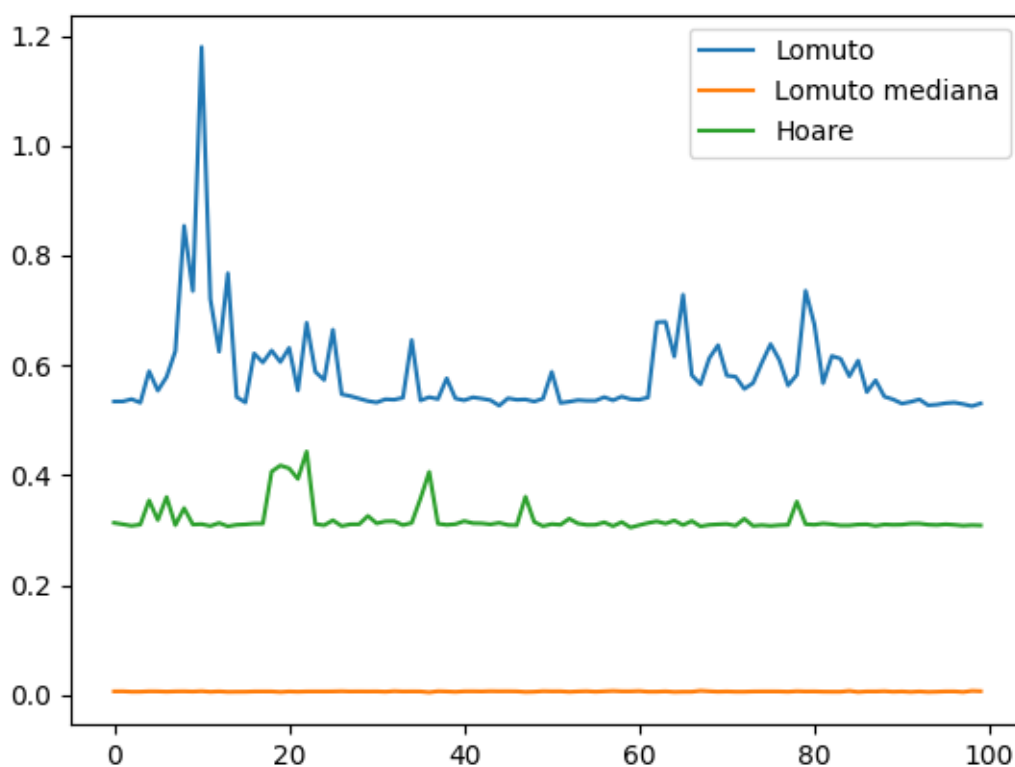
Każda wersja ma swoje lepsze i gorsze czasy, sprawdźmy jak wypadają na poszczególnych listach.

Porównanie działania na losowych listach 5000 elementowych zawierających liczby od 0 do 5.



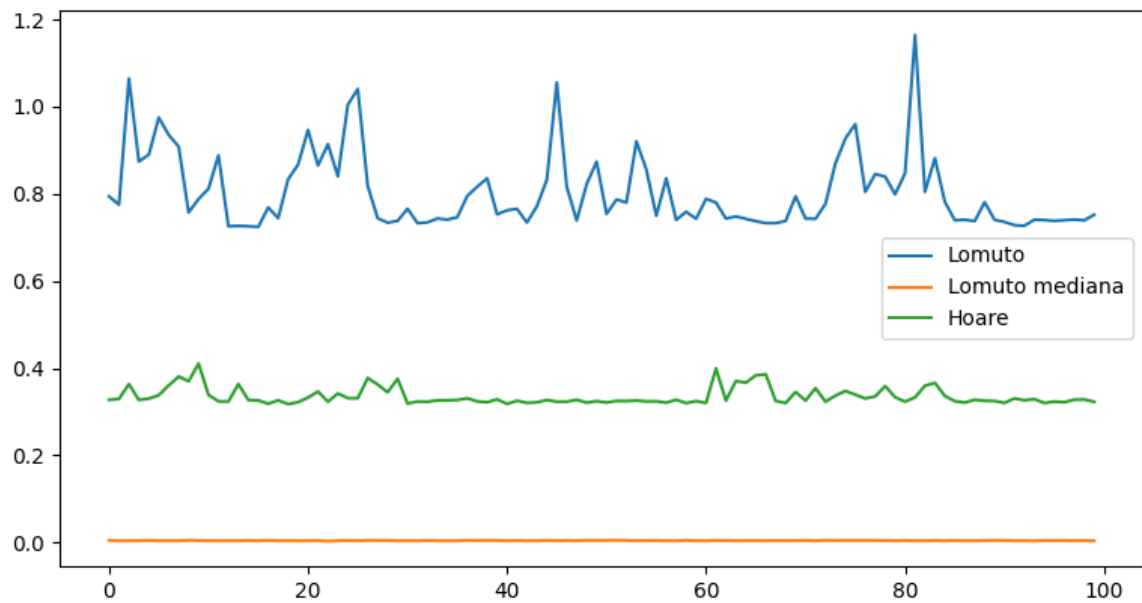
W tym wypadku najgorzej wypada QuickSort z Partition w wersji Hoare, a Wersja Lomuto i Lomuto Mediana działają w bardzo podobnym czasie.

Porównanie działania na losowych listach 5000 elementowych posortowanych malejąco.



W tym wypadku najlepiej radzi sobie QuickSort z Partition w wersji Lomuto Mediana, Zaraz po nim jest wersja Hoare i najgorzej radzi sobie wersja Lomuto.

Porównanie działania na losowych listach 1000 elementowych posortowanych rosnąco.



Tutaj sytuacja bardzo podobna co do list posortowanych malejąco.