ALGORYTMY SORTOWANIA

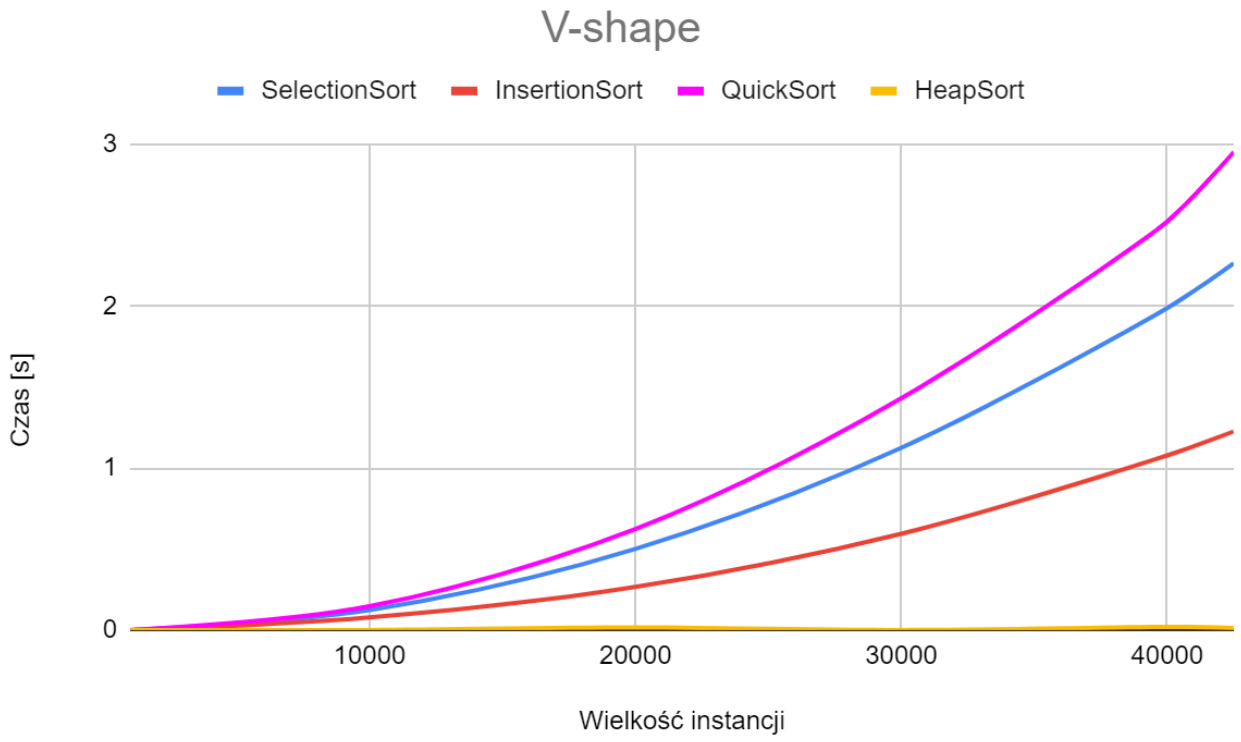
Sprawozdanie

Aleksandra Jankowska,

Aleksandra Baumgart

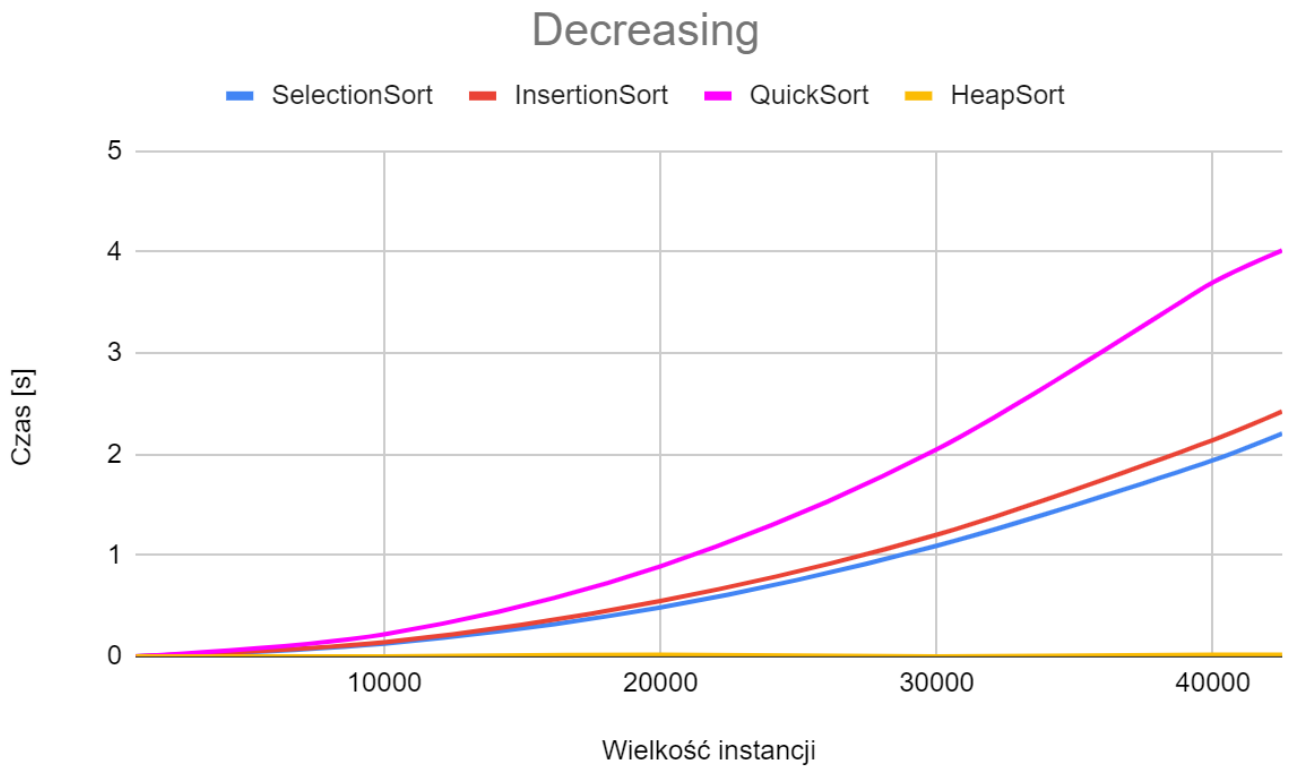
ROZKŁAD V-KSZTAŁTNY

	Czas sortowania w zależności od algorytmu			
Wielkość instancji	SelectionSort	InsertionSort	QuickSort	HeapSort
1000	0	0	0	0
10000	0,124	0,078	0,147	0
20000	0,501	0,266	0,624	0,016
30000	1,125	0,594	1,431	0
40000	1,987	1,078	2,522	0,02
42500	2,264	1,226	2,95	0,013



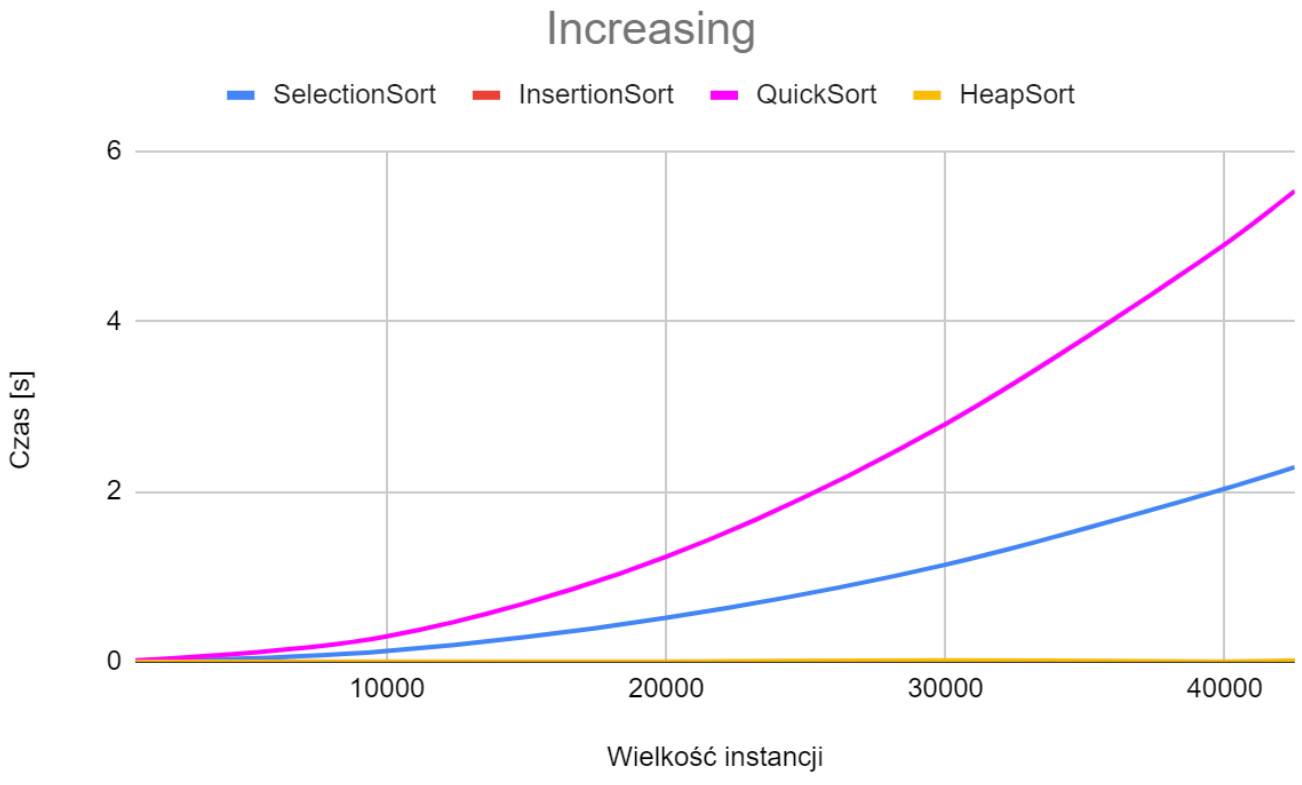
WARTOŚCI MALEJĄCE

	Czas sortowania w zależności od algorytmu			
Wielkość instancji	SelectionSort	InsertionSort	QuickSort	HeapSort
1000	0	0	0,016	0
10000	0,125	0	0,297	0
20000	0,516	0	1,234	0
30000	1,14	0	2,793	0,016
40000	2,032	0	4,903	0
42500	2,285	0	5,531	0,015



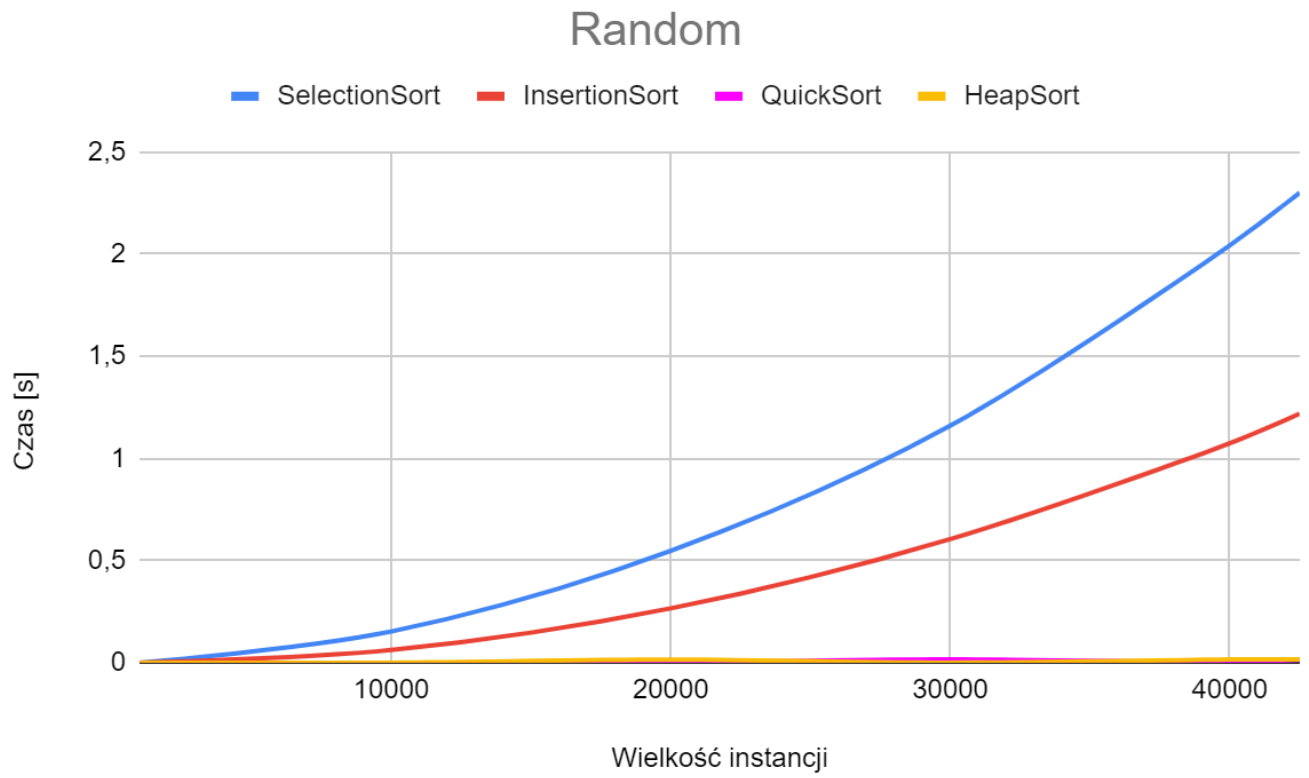
WARTOŚCI ROSNĄCE

	Czas sortowania w zależności od algorytmu			
Wielkość instancji	SelectionSort	InsertionSort	QuickSort	HeapSort
1000	0	0	0	0
10000	0,126	0,14	0,218	0
20000	0,484	0,547	0,891	0,015
30000	1,094	1,203	2,048	0
40000	1,94	2,141	3,702	0,016
42500	2,204	2,422	4,016	0,016



WARTOŚCI LOSOWE

	Czas sortowania w zależności od algorytmu			
Wielkość instancji	SelectionSort	InsertionSort	QuickSort	HeapSort
1000	0	0	0	0
10000	0,153	0,062	0	0
20000	0,548	0,266	0	0,015
30000	1,16	0,605	0,016	0
40000	2,043	1,074	0	0,016
42500	2,301	1,219	0,015	0,016



Złożoność obliczeniowa w najgorszym wypadku dla danego algorytmu

Wielkość instancji	SelectionSort	InsertionSort	QuickSort	HeapSort
10	100	100	100	33
20	400	400	400	86
30	900	900	900	147
40	1600	1600	1600	213
50	2500	2500	2500	282
60	3600	3600	3600	354
70	4900	4900	4900	429
80	6400	6400	6400	506
90	8100	8100	8100	584
100	10000	10000	10000	664

WNIOSKI

Najszybciej działającym algorytmem sortowania okazał się Heap Sort; niezależnie od wielkości instancji, dane zostały posortowane o wiele szybciej niż w przeciągu sekundy. Najwięcej czasu potrzebował algorytm sortowania Quick Sort, widać to wyraźnie przy wprowadzeniu większej ilości danych do uporządkowania.

Złożoność obliczeniowa dla Selection, Insertion i Quick Sort jest taka sama dla najgorszego przypadku – wynosi $O(n^2)$. Dla Heap Sort różni się i wynosi $O(n \log(n))$. Dla najlepszego przypadku zarówno dla Heap Sort jak i Quick Sort złożoność wynosi $O(n \log(n))$, dla Selection Sort wynosi $O(n^2)$, a dla Insertion Sort – $O(n)$. W najgorszym i najlepszym przypadku złożoność obliczeniowa jest identyczna dla Selection i Heap Sort.

ZŁOŻONOŚĆ OBLICZENIOWA ALGORYTMÓW

Algorytm	Optymistyczna złożoność obliczeniowa	Średnia złożoność obliczeniowa	Pesymistyczna złożoność obliczeniowa
Selectionsort	$O(n^2)$	$O(n^2)$	$O(n^2)$
Insertionsort	$O(n)$	$O(n^2)$	$O(n^2)$
Quicksort	$O(n \log(n))$	$O(n \log(n))$	$O(n^2)$
Heapsort	$O(n \log(n))$	$O(n \log(n))$	$O(n \log(n))$