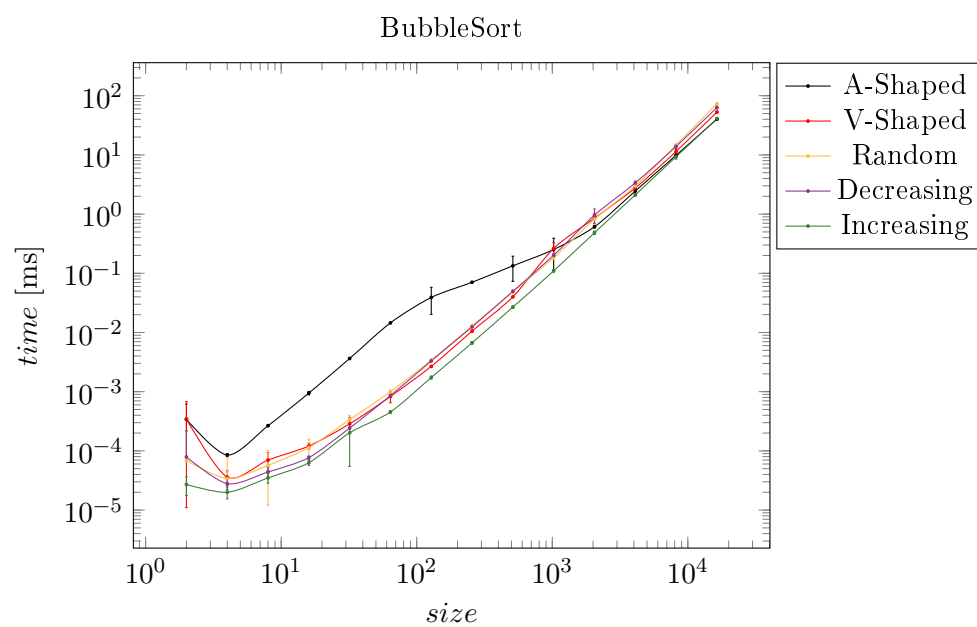


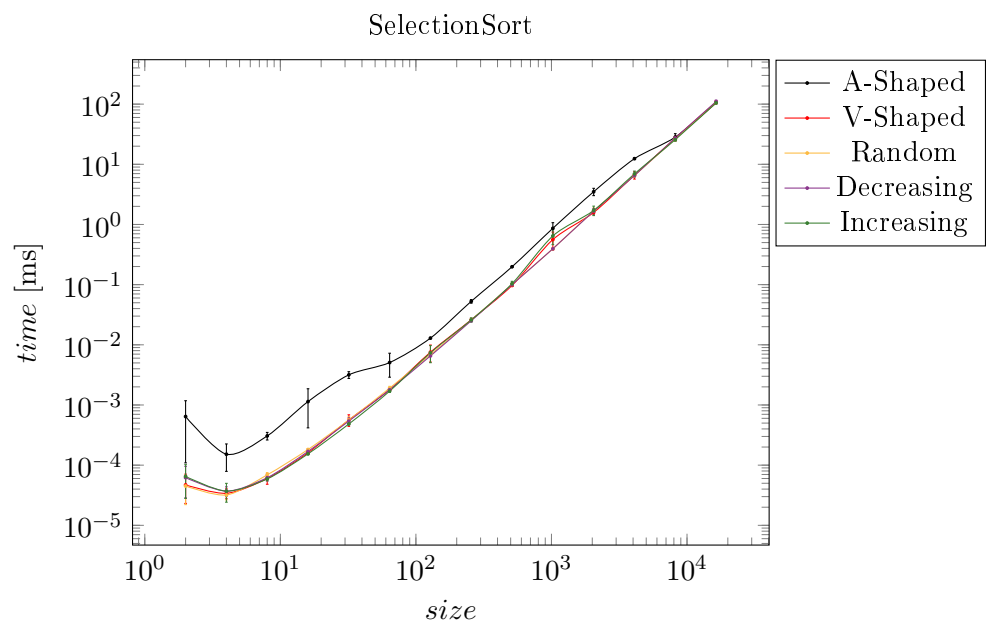
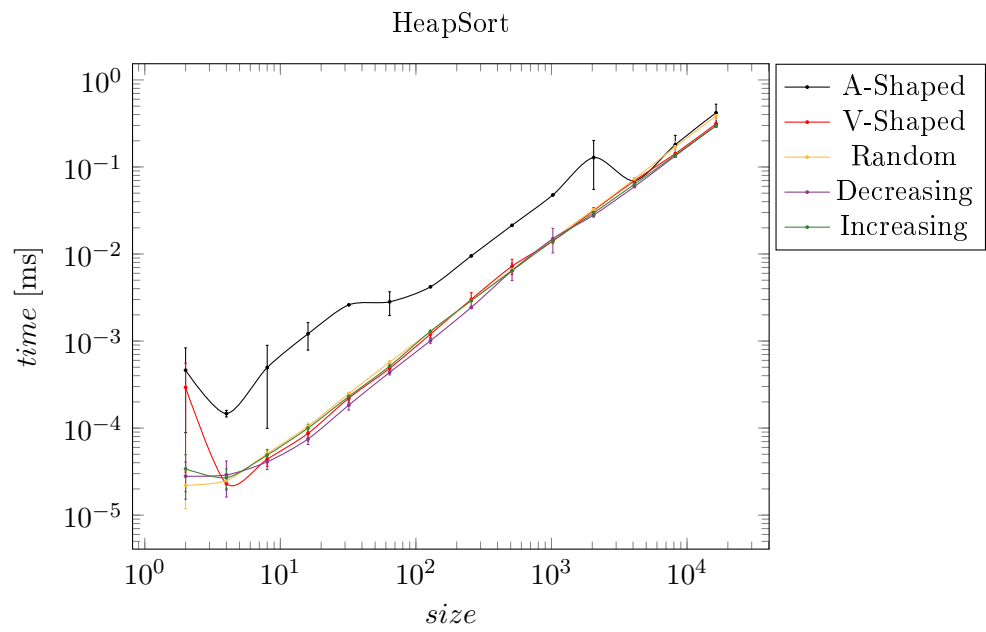
Algorytmy Sortowania

Maksymilian Ryder

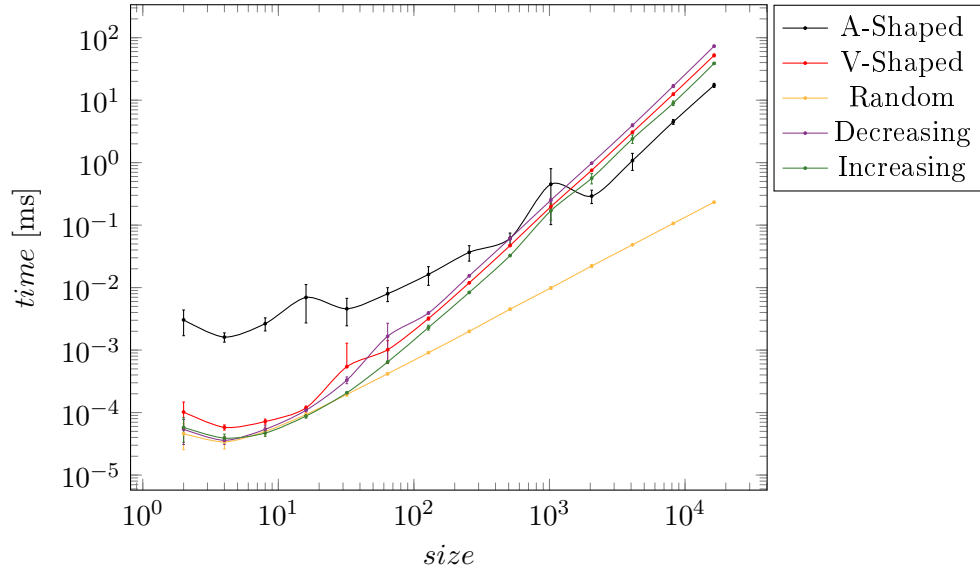
15 marca 2024

1 Zależność czasowa wybranego algorytmu od danych wejściowych

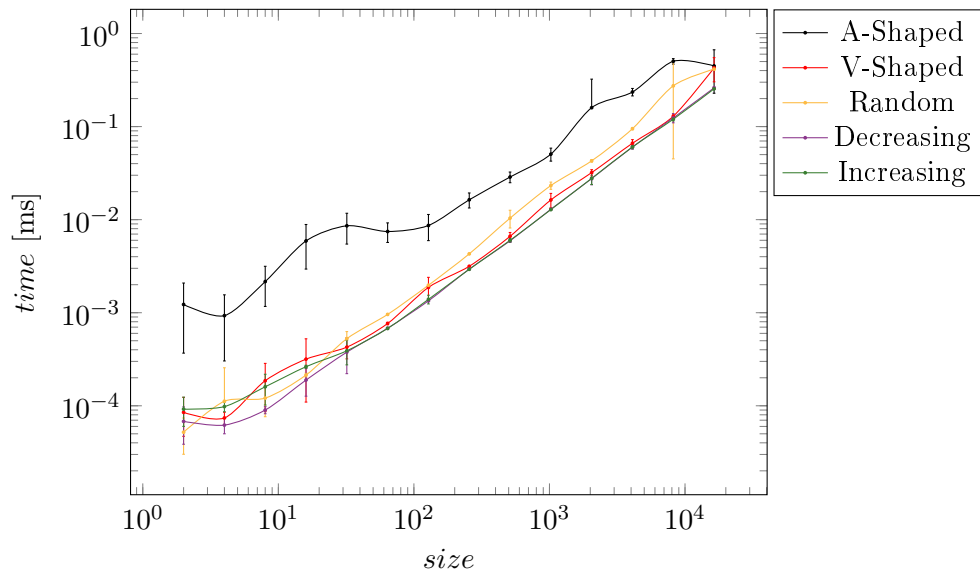


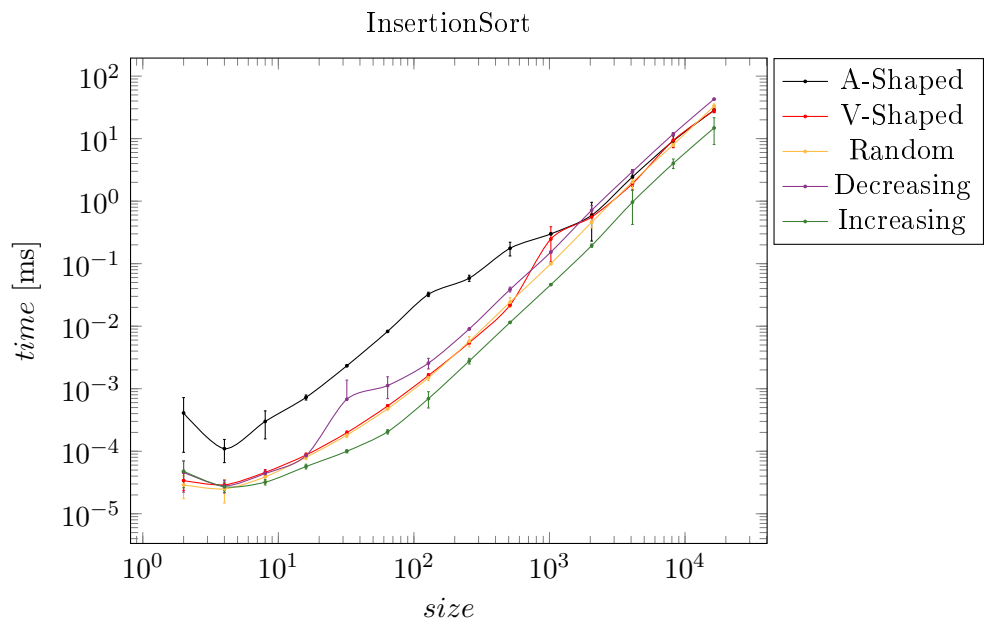


QuickSort

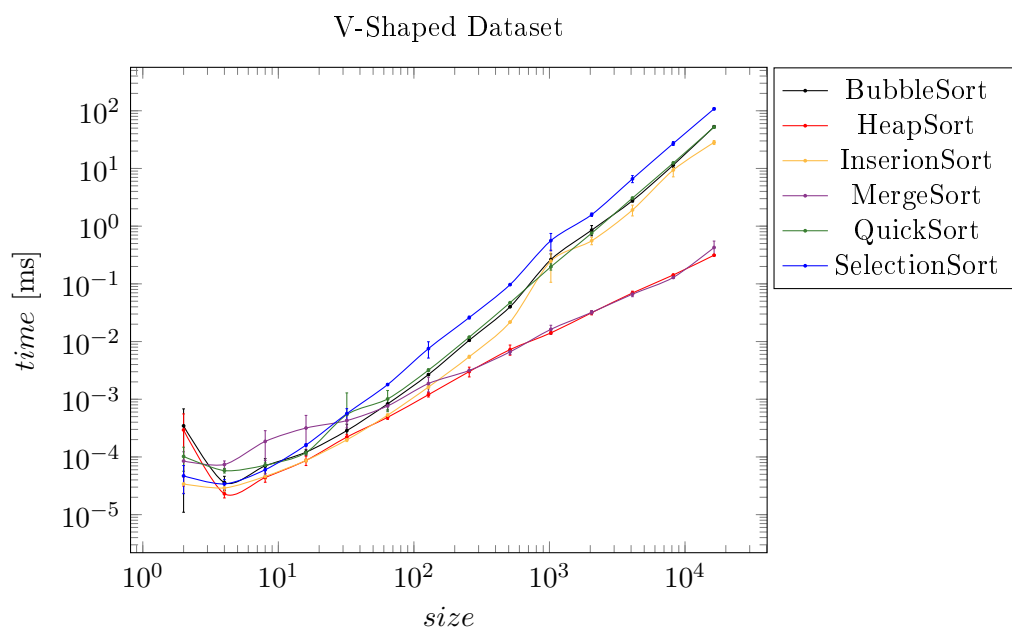
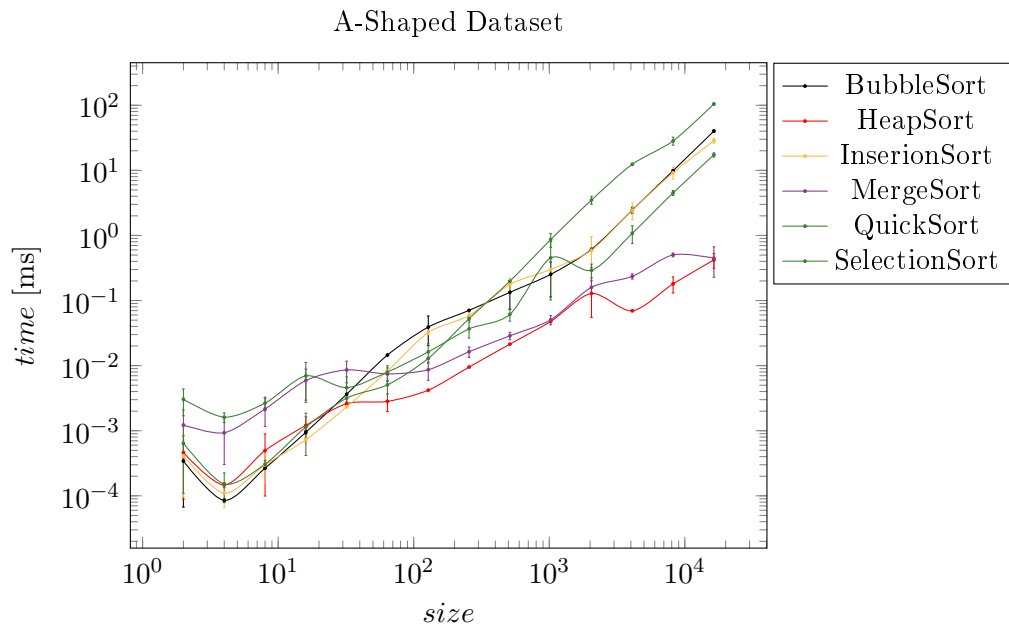


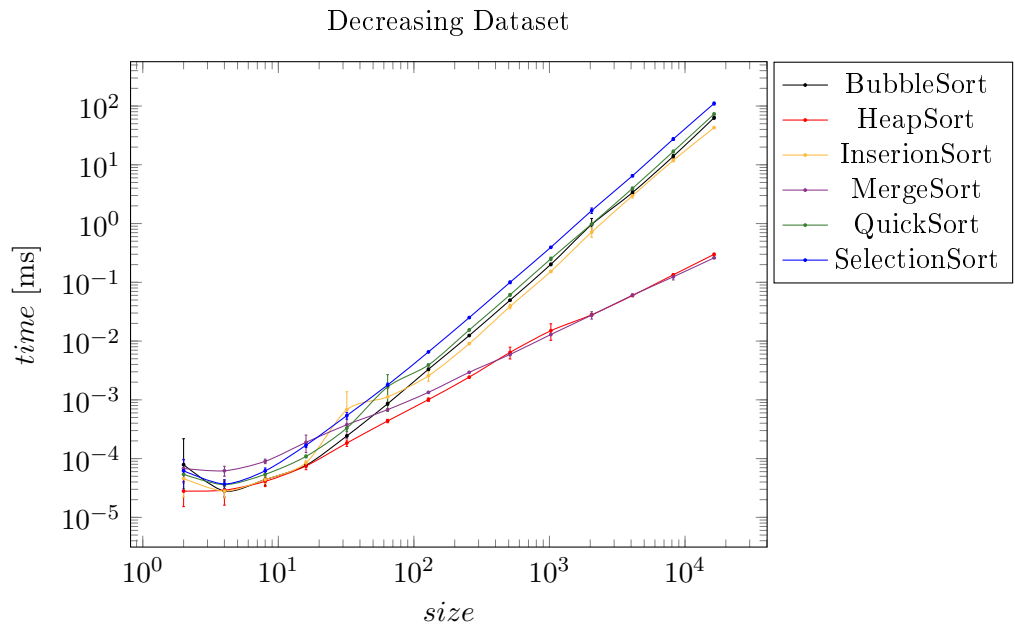
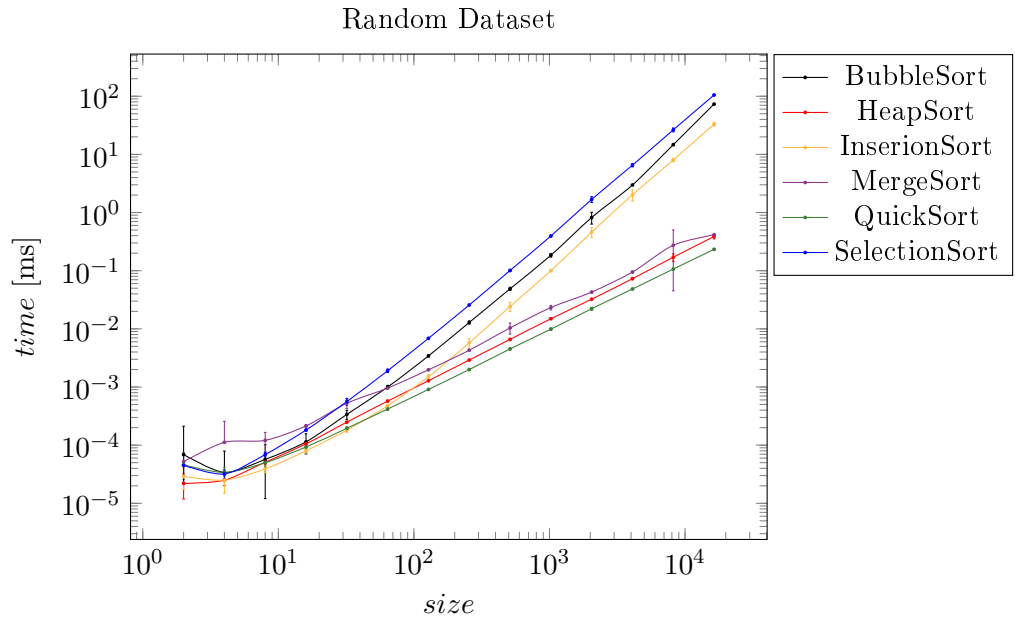
MergeSort



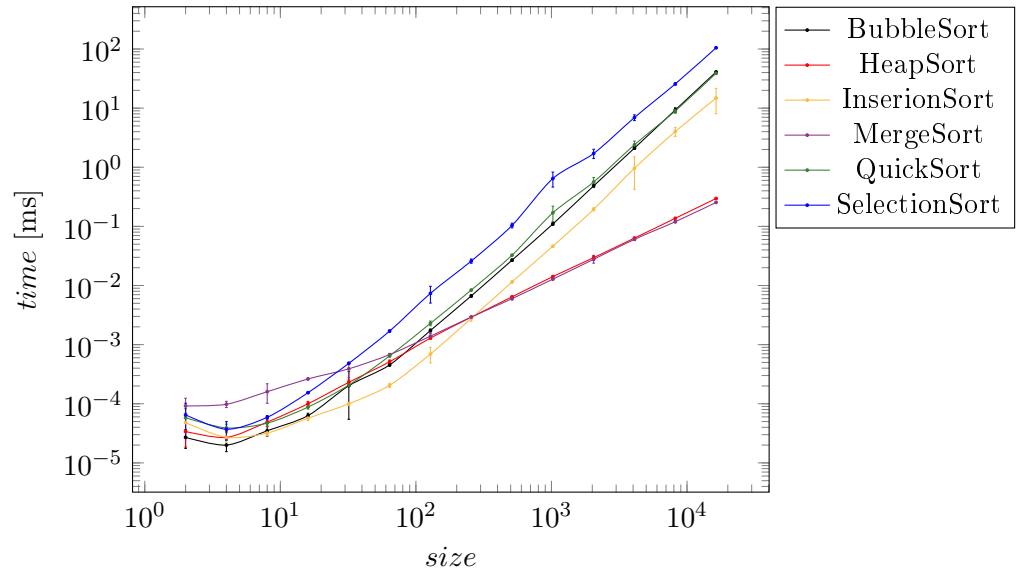


2 Efektywność czasowa zaimplementowanych metod sortowania dla wybranej postaci danych

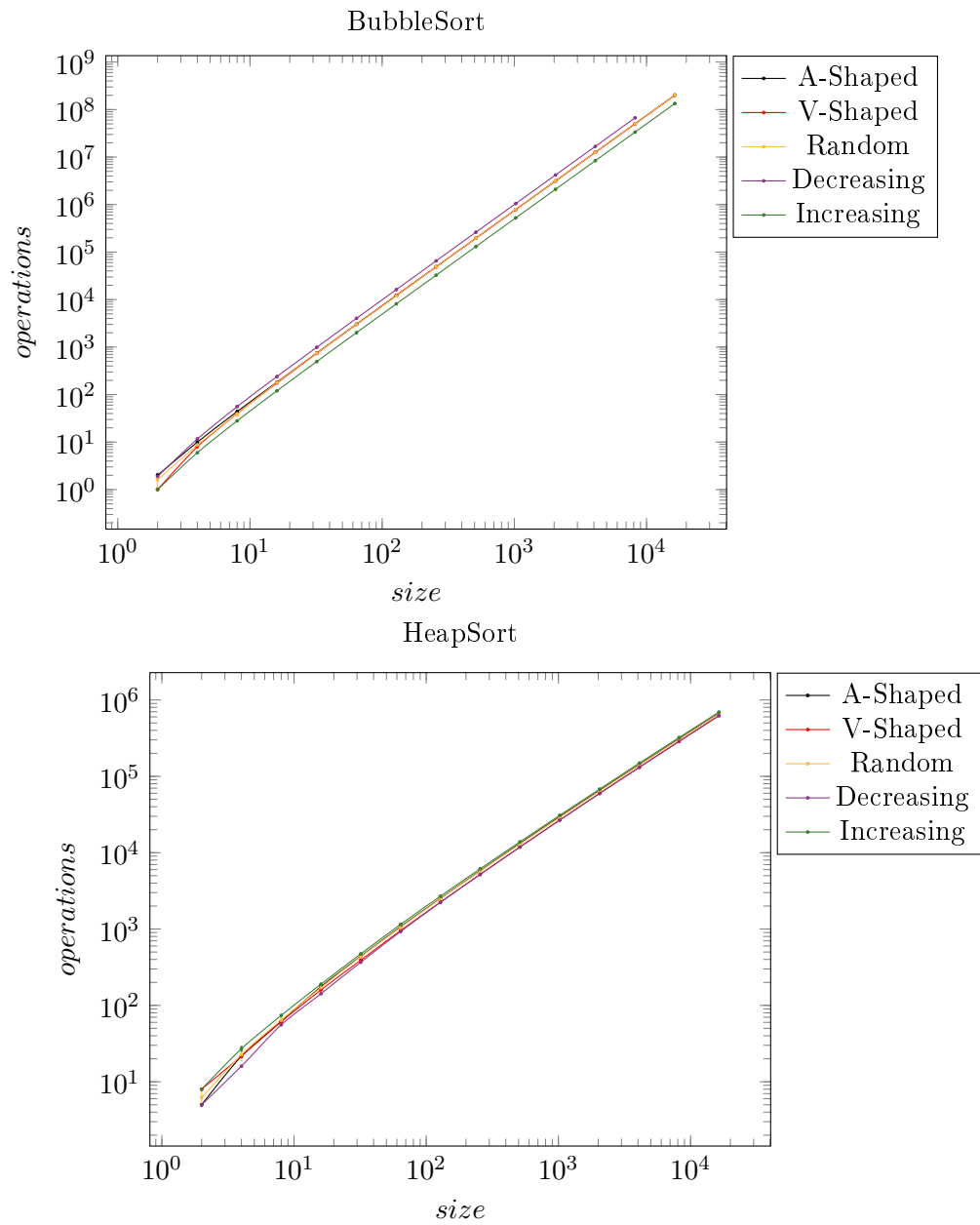


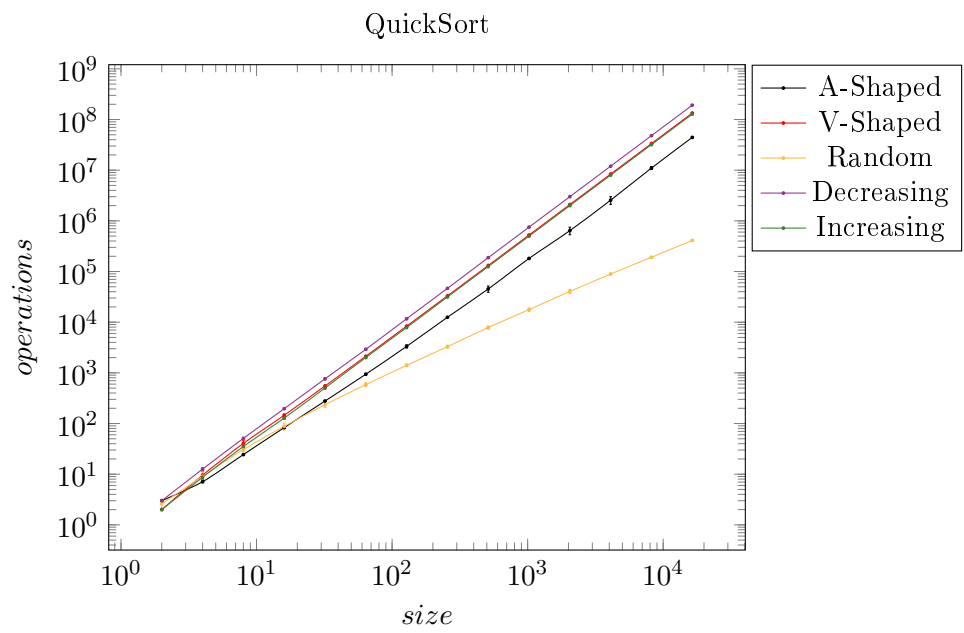
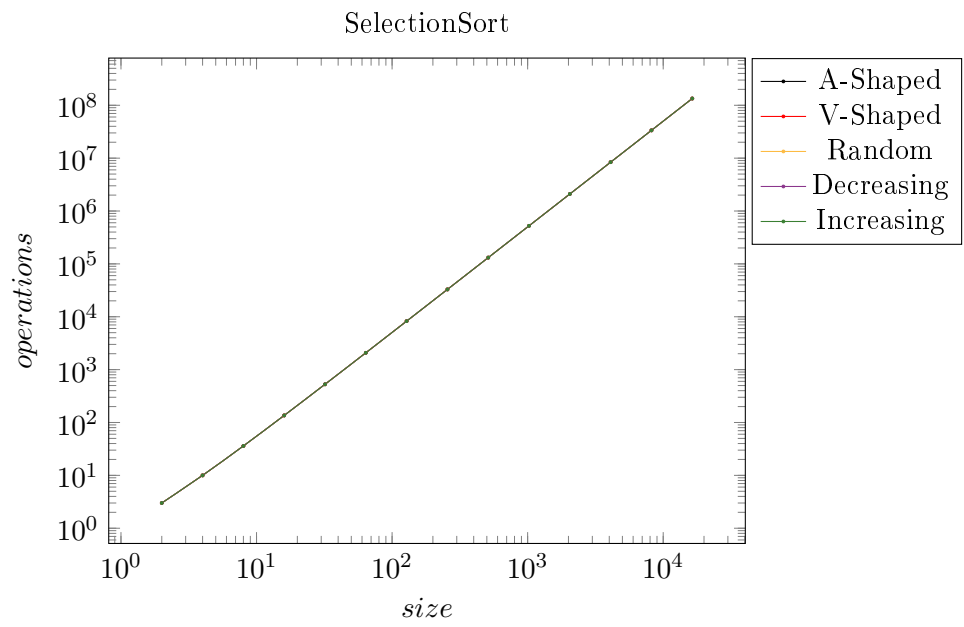


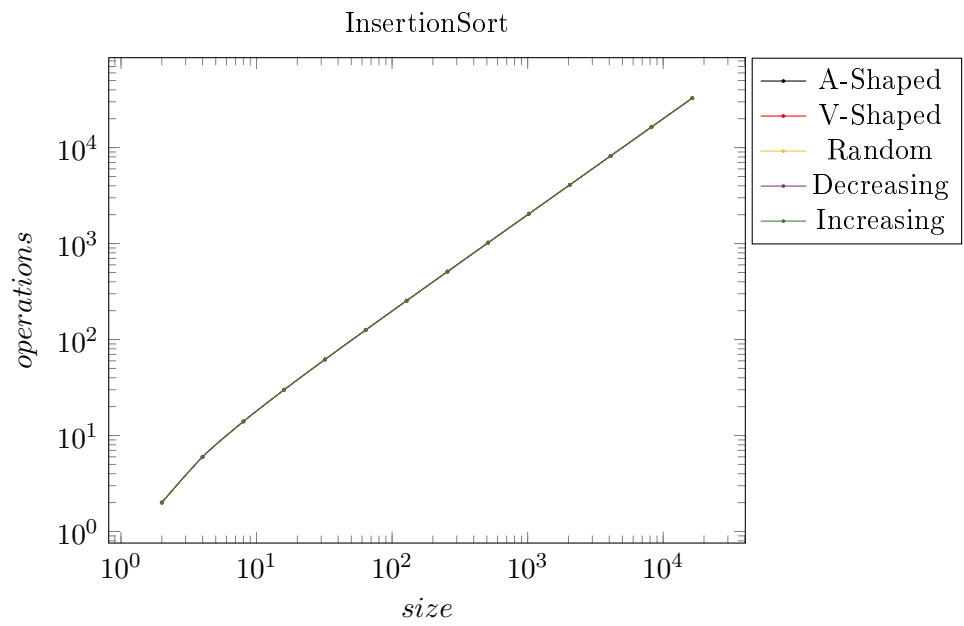
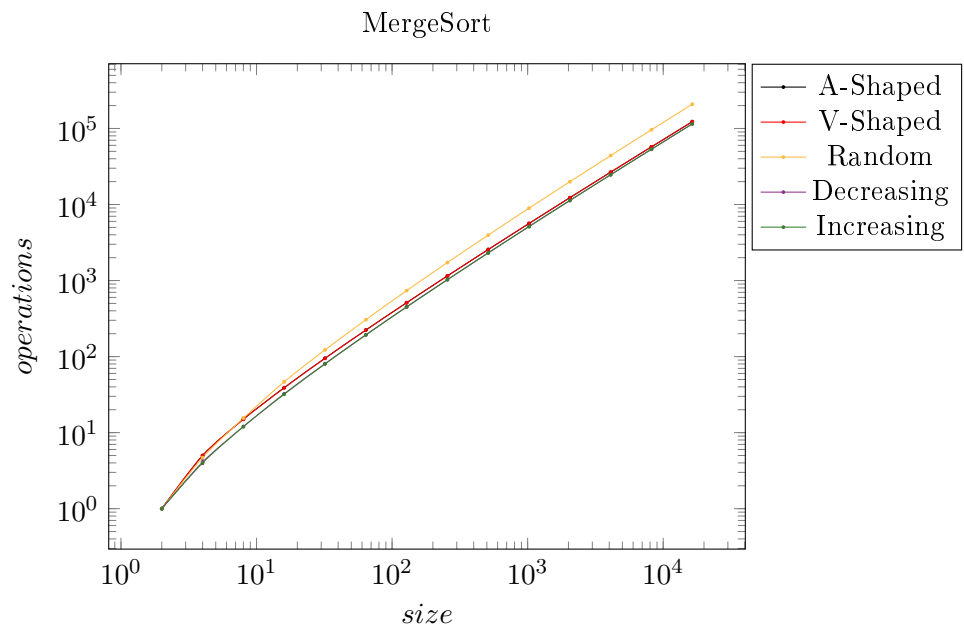
Increasing Dataset



3 Zależność operacji po wybranego algorytmu od danych wejściowych







4 Wnioski

Nazwa algorytmu	Przypadek		
	Optymistyczny	Średni	Pesymistyczny
Bubble Sort	$O(n)$	$O(n^2)$	$O(n^2)$
Insertion Sort	$O(n)$	$O(n^2)$	$O(n^2)$
Selection Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$
Heap Sort	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n \log_2 n)$
Merge Sort	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n \log_2 n)$
Quick Sort	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n^2)$

Ilość operacji wykonywanych przez algorytm jest wprost proporcjonalny do czasu działania. Widać to na wykresach, których przebiegi się pokrywają. Widać również, że algorytmy warunkowe są w stanie zapewnić lepszą złożoność w sytuacjach co najmniej optymistycznych. Algorytmy "proste", "mechaniczne" czy też po prostu "bez if'owe" mimo większej prostoty i czystości kodu przy analizie, są skazane na tą samą złożoność niezależnie od podanych danych.