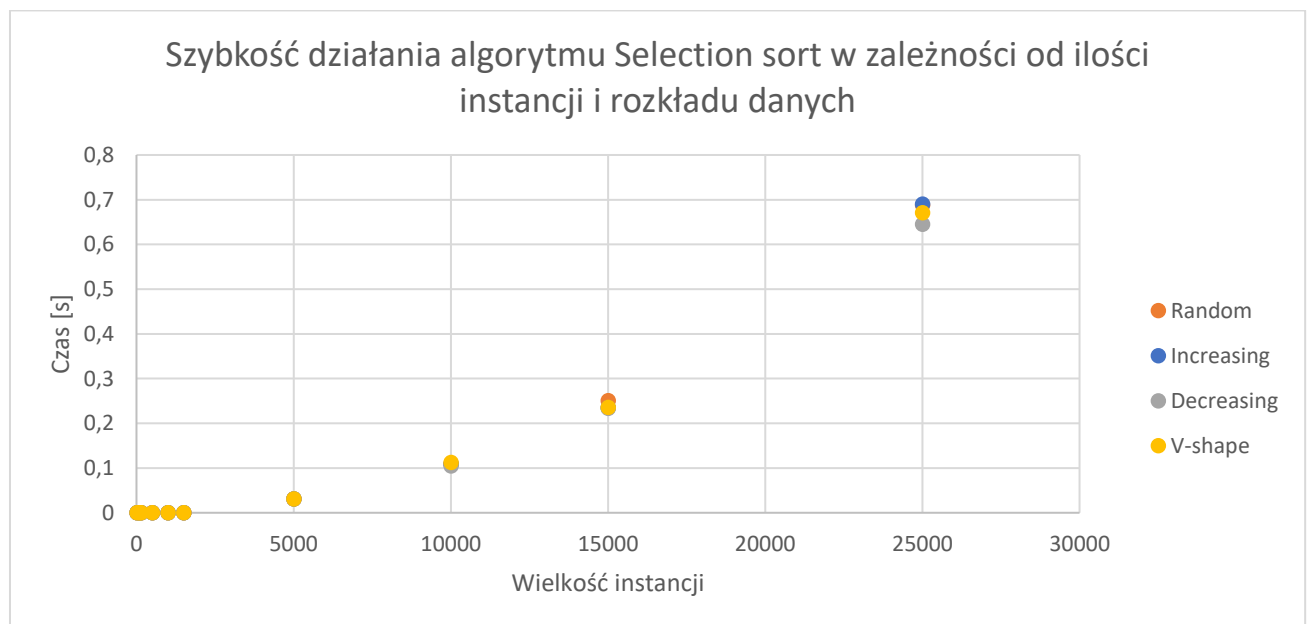


Analiza zachowania algorytmu dla różnych rozkładów danych.

Wielkość danych wejściowych instancji (10, 100, 150, 500, 1000, 1500, 5000, 10000, 15000, 25000)

Selection Sort

SELECTION SORT				
Dane uporządkowane	Losowo	rosnąco	malejąco	V-shape
WIELKOŚĆ INSTANCJI	CZAS[s]			
10	0	0	0	0
100	0	0	0	0
150	0	0	0	0
500	0	0	0	0
1000	0	0	0	0
1500	0	0	0	0
5000	0,031	0,031	0,031	0,031
10000	0,109	0,109	0,105	0,113
15000	0,251	0,235	0,234	0,236
25000	0,688	0,69	0,645	0,671



Czas pracy algorytmu Selection Sort dla podanych wielkości instancji w zależności od rodzaju rozłożenia danych.

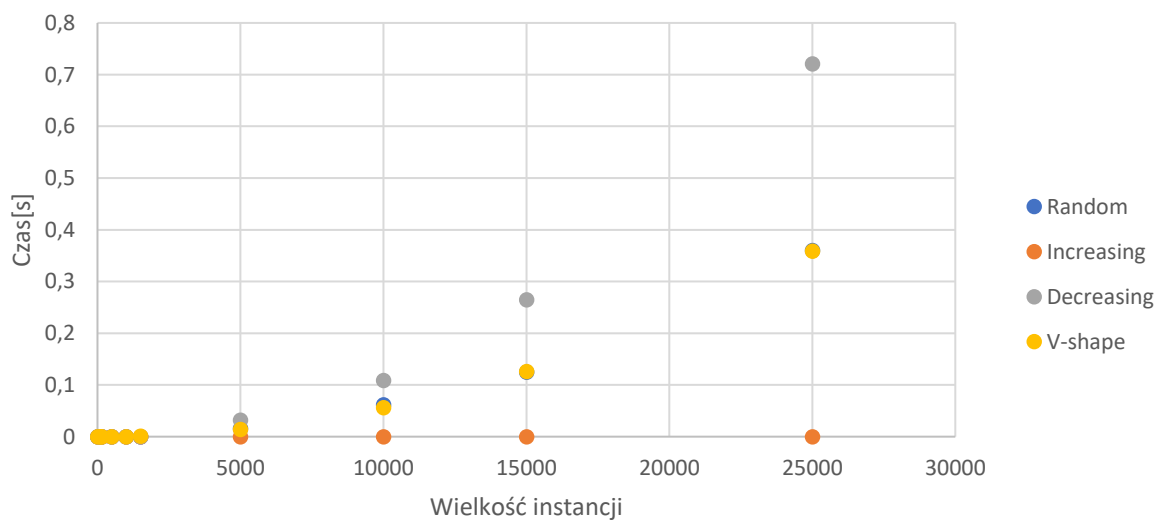
Decreasing < V-shape < Random < Increasing

Złożoność obliczeniowa $O(n^2)$ dla wszystkich przypadków.

Insertion Sort

INSERTION SORT				
Dane uporządkowane	Losowo	rosnąco	malejąco	V-shape
WIELKOŚĆ INSTANCJI	Czas[s]			
10	0	0	0	0
100	0	0	0	0
150	0	0	0	0
500	0	0	0	0
1000	0	0	0	0
1500	0	0	0	0,001
5000	0,016	0	0,032	0,014
10000	0,062	0	0,109	0,056
15000	0,125	0	0,265	0,126
25000	0,36	0	0,721	0,359

Szybkość działania algorytmu Insertion sort w zależności od ilości instancji i rozkładu danych



Czas pracy algorytmu Insertion Sort dla podanych wielkości instancji w zależności od rodzaju rozłożenia danych.

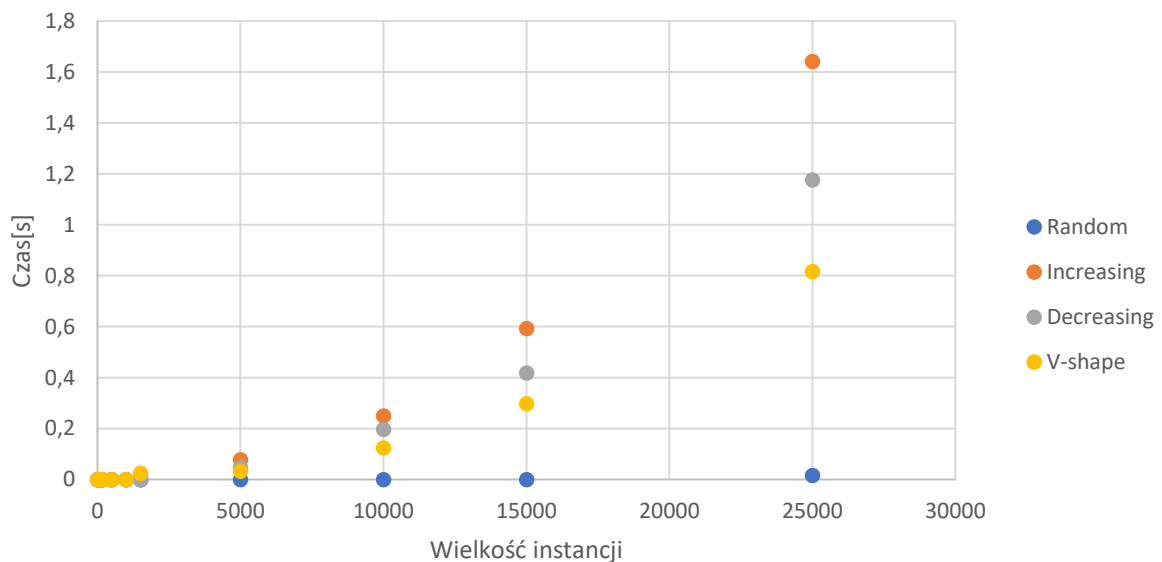
Increasing < V-shape < Random < Decreasing

Złożoność obliczeniowa $O(n^2)$ dla wszystkich przypadków.

Quick Sort

QUICK SORT				
Dane uporządkowane	Losowo	rosnąco	malejąco	V-shape
WIELKOŚĆ INSTANCJI	CZAS[s]			
10	0	0	0	0
100	0	0	0	0
150	0	0	0	0
500	0	0	0	0
1000	0	0	0	0
1500	0	0	0	0,025
5000	0	0,078	0,047	0,032
10000	0	0,25	0,197	0,124
15000	0	0,593	0,418	0,297
25000	0,016	1,641	1,176	0,816

Szybkość działania algorytmu Quick sort w zależności od ilości instancji i rozkładu danych



Czas pracy algorytmu Quick Sort dla podanych wielkości instancji w zależności od rodzaju rozłożenia danych.

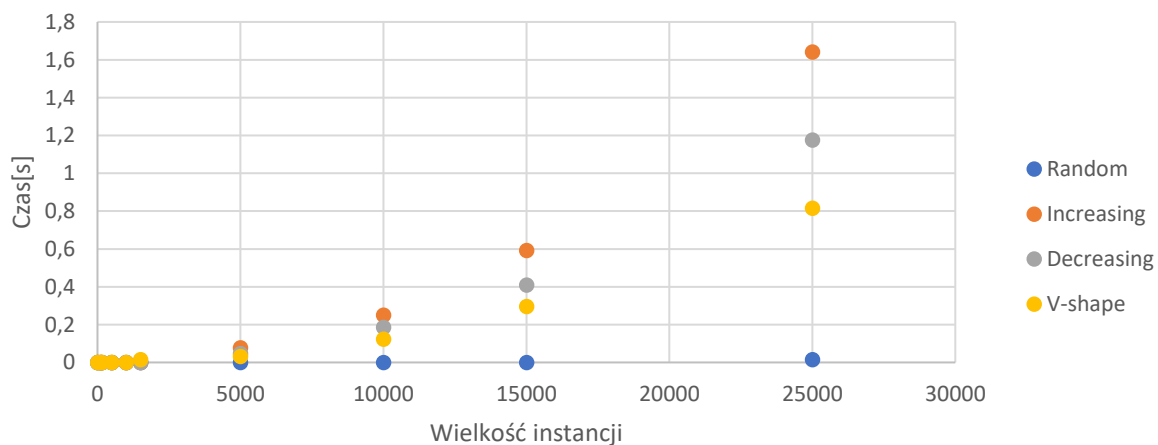
Random < V-shape < Decreasing < Increasing

Złożoność obliczeniowa $O(n^2)$ dla najgorszego przypadku a $O(n \cdot \log n)$ dla najlepszego przypadku.

Heap Sort

HEAP SORT				
Dane uporządkowane	Losowo	rosnąco	malejąco	V-shape
WIELKOŚĆ INSTANCJI	CZAS[s]			
10	0	0	0	0
100	0	0	0	0
150	0	0	0	0
500	0	0	0	0
1000	0	0	0	0
1500	0	0	0	0,015
5000	0	0,078	0,047	0,032
10000	0	0,25	0,187	0,124
15000	0	0,593	0,41	0,297
25000	0,016	1,641	1,176	0,816

Szybkość działania algorytmu Heap sort w zależności od ilości instancji i rozkładu danych



Czas pracy algorytmu Heap Sort dla podanych wielkości instancji w zależności od rodzaju rozłożenia danych.

Random < V-shape < Decreasing < Increasing

Złożoność obliczeniowa $O(n)$ dla najlepszego przypadku a $O(n \cdot \log n)$ dla najgorszego przypadku.

Wnioski:

Na podstawie analizy czasów pracy poszczególnych algorytmów można stwierdzić, że najkorzystniejszą opcją jest losowe rozłożenie danych. Natomiast tą najmniej korzystną jest rozłożenie danych w sposób rosnący (increasing). Po porównaniu czasu pracy i złożoności obliczeniowej algorytmów można wysunąć wniosek, że najlepszym algorytmem sortowania jest Heap Sort ponieważ działa on zdecydowanie najszybciej, co szczególnie można zauważyć przy dużych wartościach.