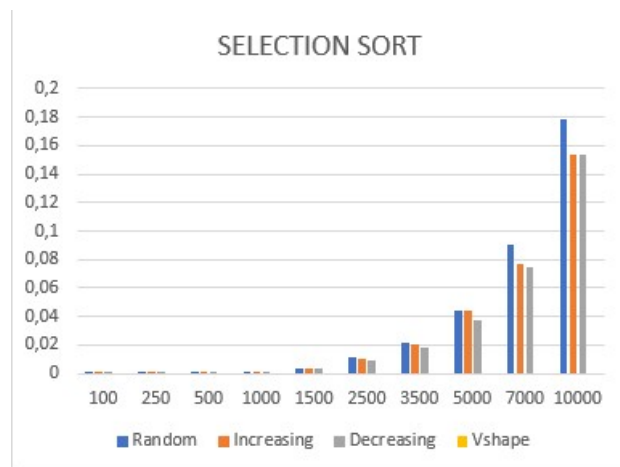


SPRAWOZDANIE: ALGORYTMY SORTUJĄCE

SELECTION SORT

Wielkość instancji / typ danych	100	250	500	1000	1500	2500	3500	5000	7000	10000
Random	0,000025	0,000128	0,000463	0,001668	0,003727	0,011282	0,021919	0,043556	0,0901276	0,178446
Increasing	0,000023	0,000110	0,000459	0,001729	0,003817	0,010348	0,020931	0,043733	0,076430	0,153265
Decreasing	0,000019	0,000119	0,000470	0,001561	0,003673	0,008903	0,018587	0,037693	0,074636	0,153301
Vshape										

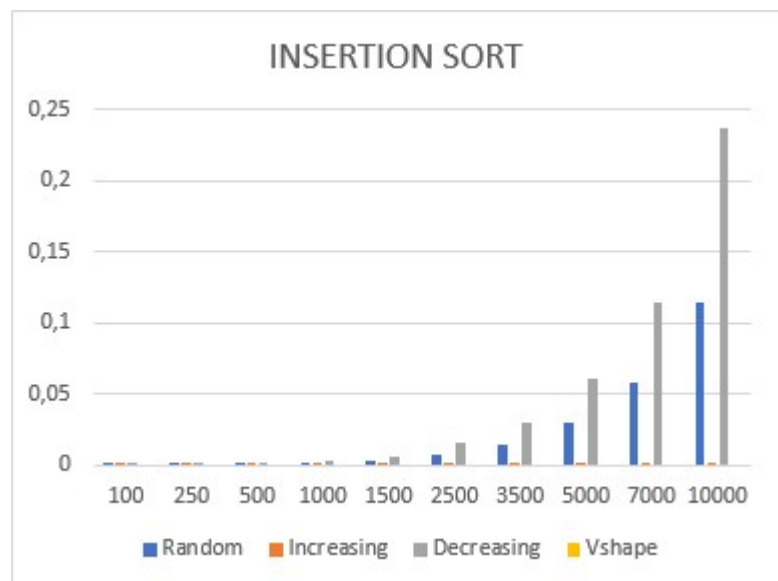


Identyfikacja przypadków: najlepszy, średni oraz najgorszy przypadek to $O(n^2)$. Jest to sortowanie w miejscu.

Sortowanie typu selection sort jest stosunkowo tak samo szybkie dla typów danych: increasing oraz decreasing. Jest wolniejsze dla wartości randomowych. Znaczącą różnicę w czasie można zobaczyć pomiędzy liczbą instancji 1500 a 2500.

INSERTION SORT

Wielkość instancji / typ danych	100	250	500	1000	1500	2500	3500	5000	7000	10000
Random	0,000014	0,000074	0,000364	0,001178	0,002676	0,007234	0,014654	0,030402	0,057835	0,114612
Increasing	0,000002	0,000003	0,000004	0,000008	0,000010	0,000015	0,000024	0,000034	0,000047	0,000064
Decreasing	0,000024	0,000138	0,000552	0,002402	0,005352	0,015364	0,029291	0,060902	0,114546	0,237769
Vshape										

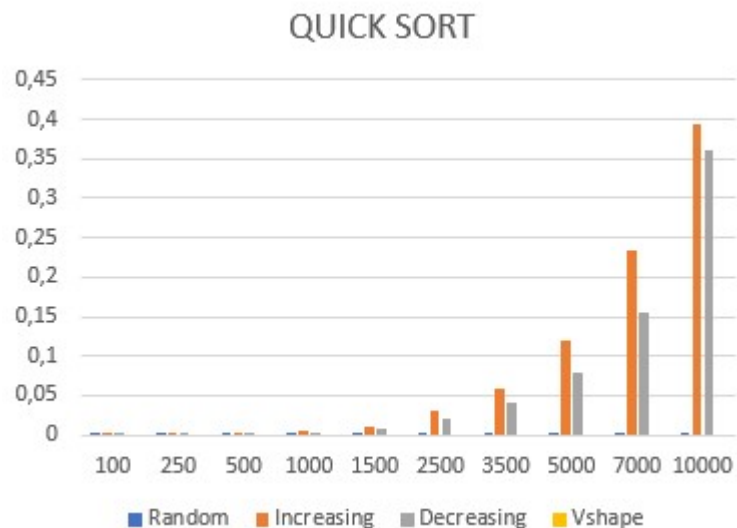


Identyfikacja przypadków: najlepszy - $O(n)$, średni - $O(n \log n)$, najgorszy - $O(n^2)$. Jest to sortowanie w miejscu.

Insertion Sort najszybciej sortuje liczby od najmniejszych do największych, następnie randomowe, potem liczby malejące - decreasing. Pomiędzy liczbami randomowymi a malejącymi czas zwiększa się o połowę.

QUICK SORT

Wielkość instancji / typ danych	100	250	500	1000	1500	2500	3500	5000	7000	10000
Random	0,000010	0,000028	0,000066	0,000149	0,000218	0,000399	0,000609	0,000889	0,001245	0,002203
Increasing	0,000048	0,000301	0,001159	0,004626	0,010491	0,029704	0,058238	0,119423	0,233144	0,394123
Decreasing	0,000049	0,000296	0,001102	0,003968	0,008445	0,021010	0,039966	0,079047	0,155199	0,359741
Vshape										



Identyfikacja przypadków: najlepszy – $O(n \log n)$, średni – $O(n \log n)$, najgorszy – $O(n^2)$. Jest to sortowanie w miejscu.

Quick Sort najszybciej sortuje liczby randomowe, następnie rosnące, potem malejące.

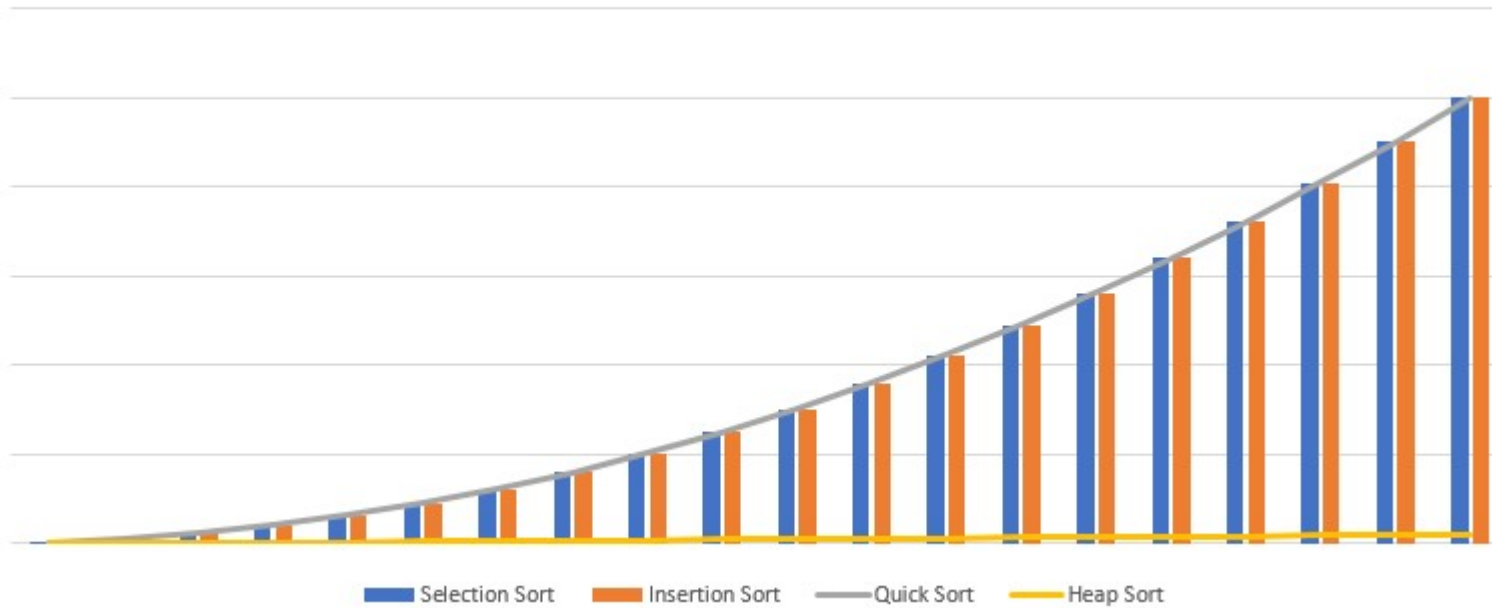
HEAP SORT

Wielkość instancji / typ danych	100	250	500	1000	1500	2500	3500	5000	7000	10000
Random										
Increasing										
Decreasing										
Vshape										

Identyfikacja przypadków: najgorszy – $O(n \log n)$. Jest to sortowanie w miejscu.

ZŁOŻONOŚĆ OBLICZENIOWA DLA NAJGORSZEGO PRZYPADKU

Złożoność: najgorszy przypadek



Złożoność obliczeniowa dla najgorszego przypadku wynosi:

selection sort, insertion sort, quick sort – $O(n^2)$

heap sort – $O(n \log n)$

