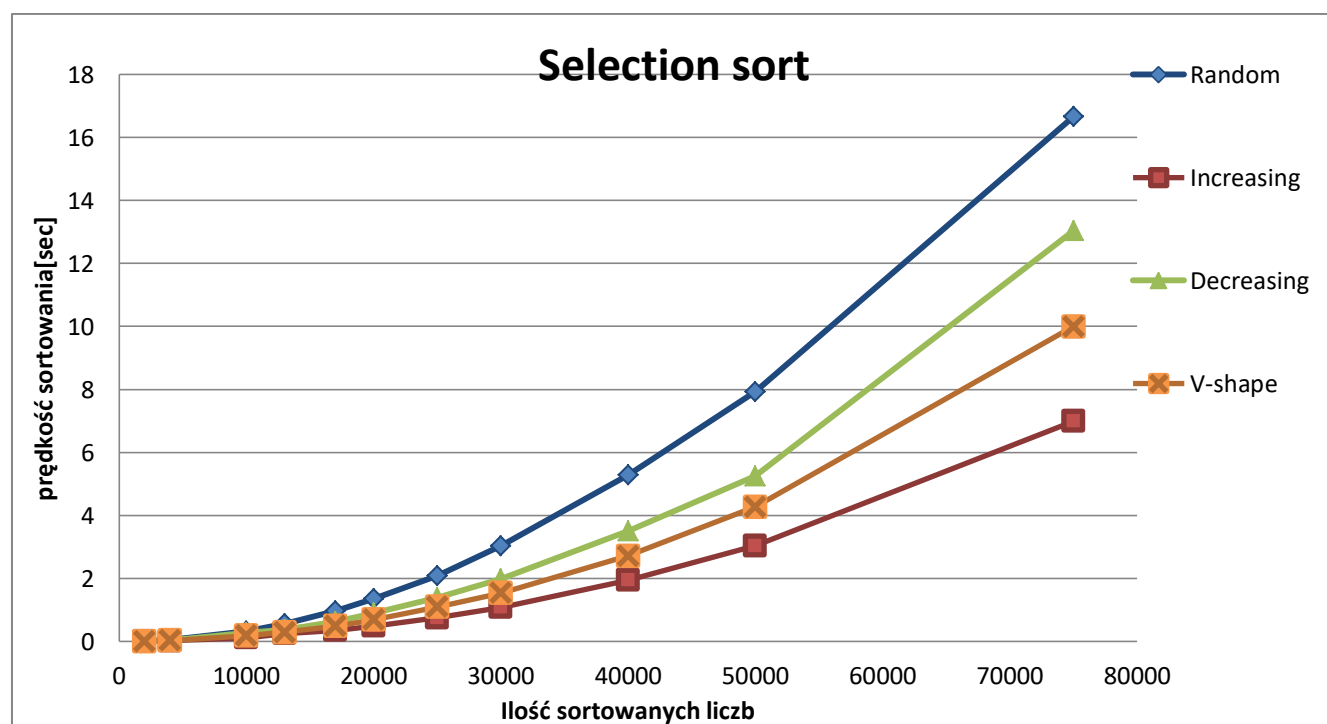


Sprawozdanie projektu dot. Algorytmów sortowania.

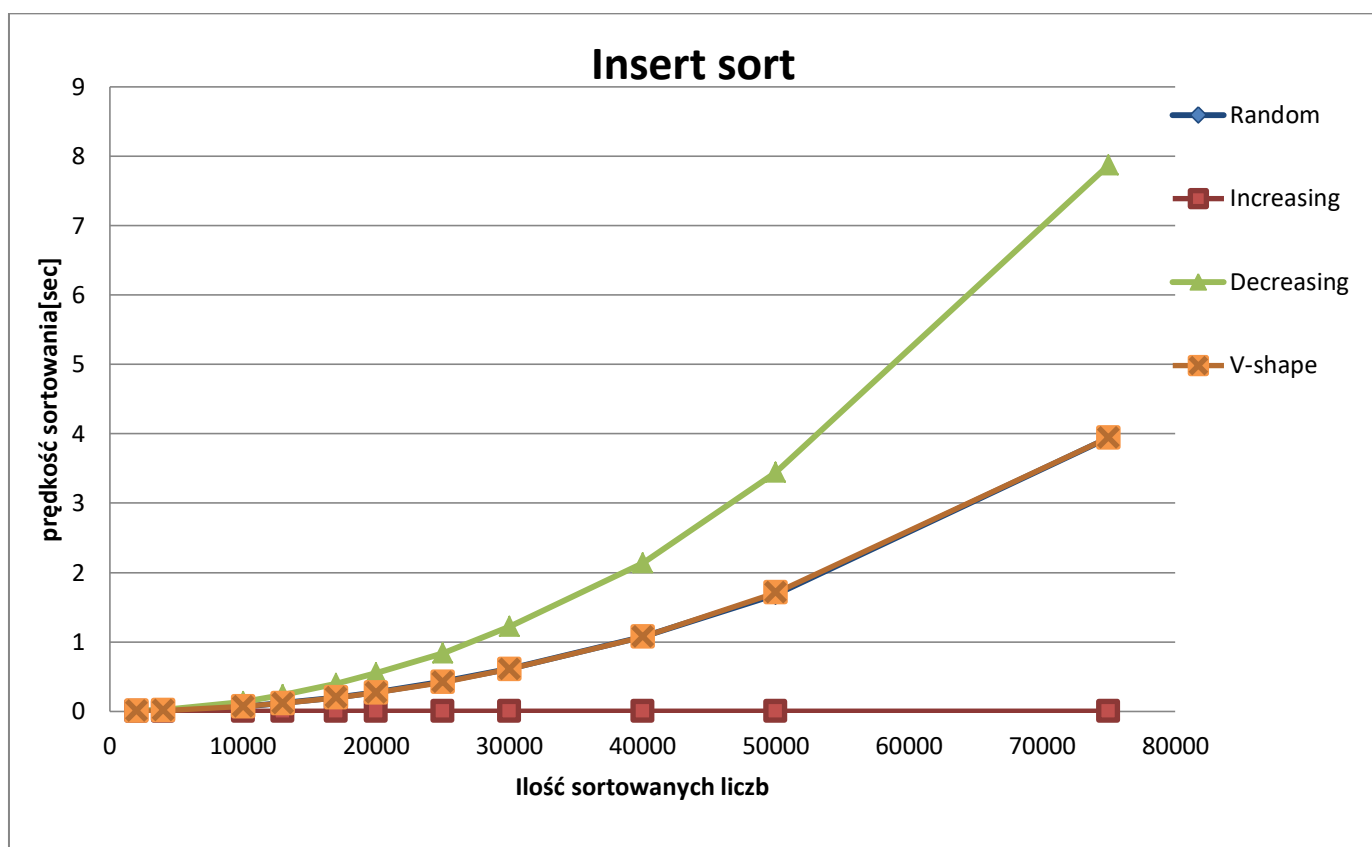
Mikołaj Smolibowski 145173

Wojciech Łosiewski 145187

Selection sort	Typ wprowadzanych danych			
Sortowane numery	Random	Increasing	Decreasing	V-shape
2000	0,009	0,005	0,009	0,007
4000	0,045	0,02	0,036	0,028
10000	0,317	0,119	0,27	0,184
13000	0,56	0,254	0,382	0,294
17000	0,975	0,355	0,636	0,492
20000	1,362	0,488	0,897	0,692
25000	2,089	0,749	1,39	1,095
30000	3,03	1,08	1,99	1,536
40000	5,286	1,948	3,513	2,719
50000	7,92	3,037	5,253	4,271
75000	16,667	6,986	13,041	9,992

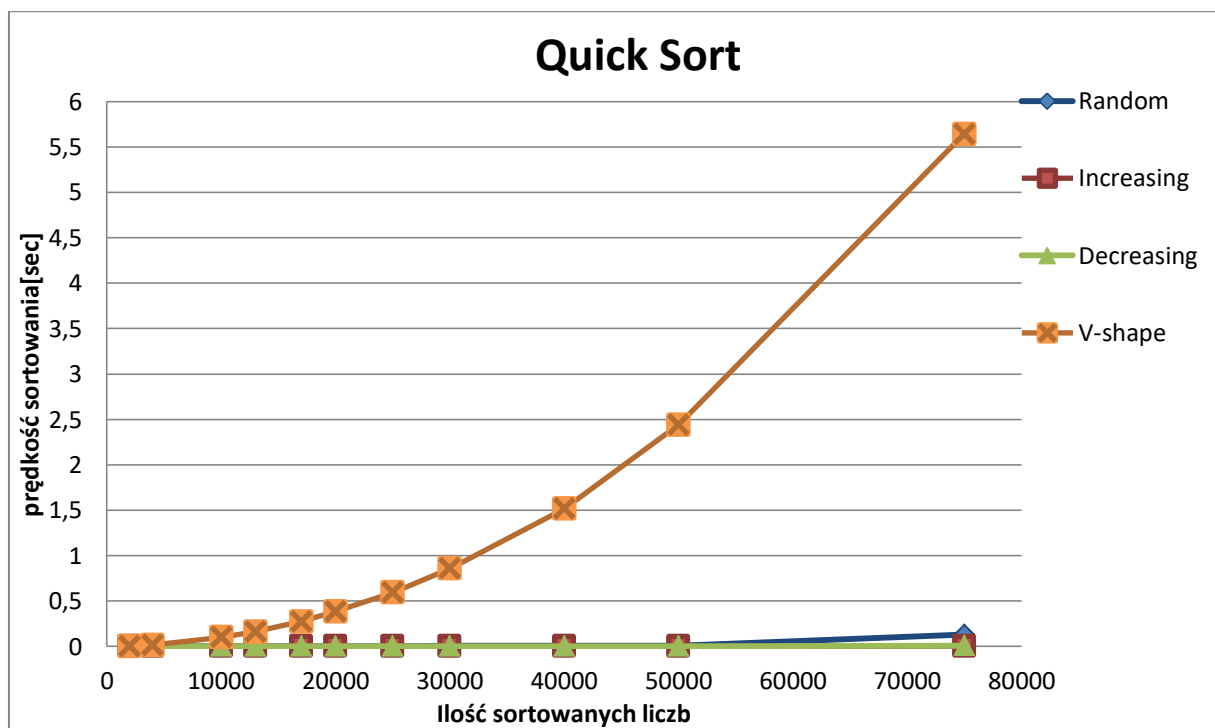


Insert sort	Typ wprowadzanych danych			
Sortowane numery	Random	Increasing	Decreasing	V-shape
2000	0,003	0	0,006	0,002
4000	0,011	0	0,021	0,012
10000	0,068	0	0,139	0,069
13000	0,117	0	0,232	0,114
17000	0,194	0	0,398	0,195
20000	0,277	0	0,551	0,267
25000	0,43	0	0,834	0,417
30000	0,611	0	1,223	0,607
40000	1,07	0	2,138	1,072
50000	1,678	0	3,445	1,708
75000	3,94	0	7,872	3,944

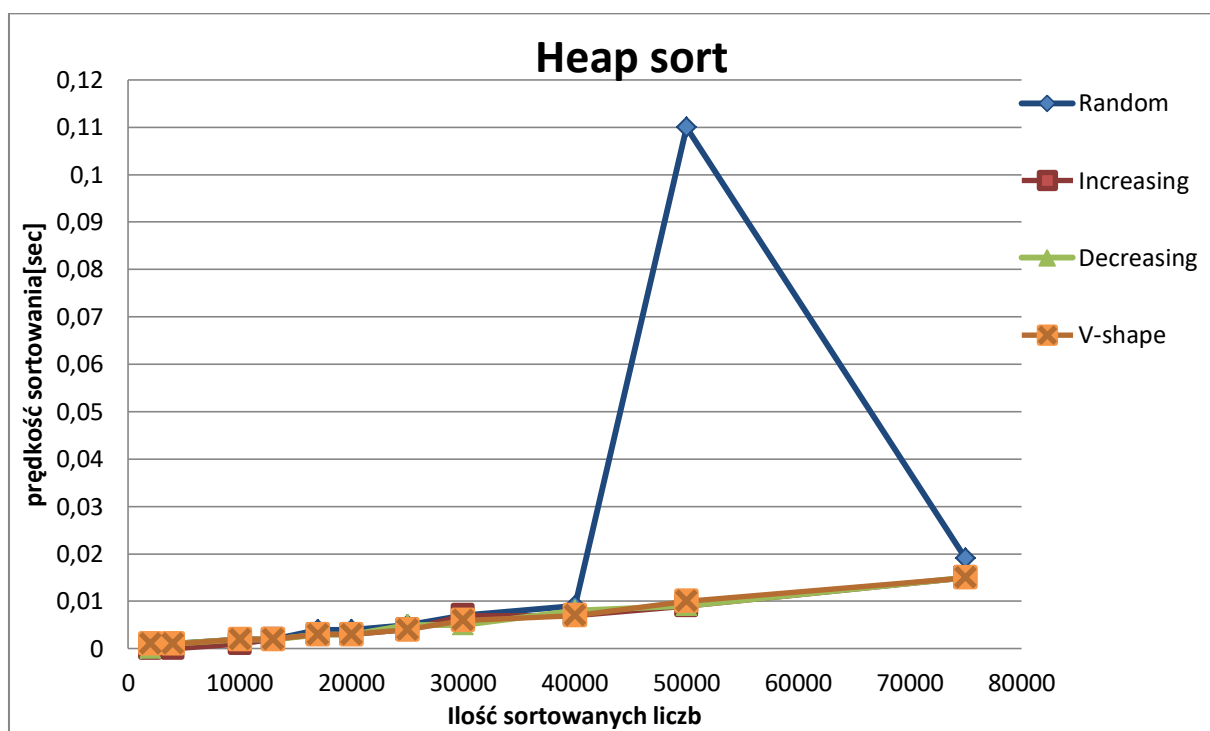


(W wykresie Insert sort, linie Random u V-shape nakładają się na siebie)

Quick sort	Typ wprowadzanych danych			
Sortowane numery	Random	Increasing	Decreasing	V-shape
2000	0,001	0,001	0	0,004
4000	0	0,001	0	0,016
10000	0,002	0	0	0,099
13000	0,002	0,001	0,001	0,162
17000	0,003	0,001	0,001	0,274
20000	0,003	0,001	0,001	0,384
25000	0,004	0,001	0,002	0,595
30000	0,005	0,001	0,002	0,858
40000	0,006	0,001	0,003	1,516
50000	0,008	0,002	0,004	2,44
75000	0,13	0,003	0,007	5,641



Heap sort	Typ wprowadzanych danych			
Sortowane numery	Random	Increasing	Decreasing	V-shape
2000	0,001	0	0	0,001
4000	0,001	0	0,001	0,001
10000	0,002	0,001	0,002	0,002
13000	0,002	0,002	0,002	0,002
17000	0,004	0,003	0,003	0,003
20000	0,004	0,003	0,003	0,003
25000	0,005	0,004	0,005	0,004
30000	0,007	0,007	0,005	0,006
40000	0,009	0,007	0,008	0,007
50000	0,11	0,009	0,009	0,01
75000	0,019	0,015	0,015	0,015



Omówienie wyników i wnioski.

W porównaniu czterech różnych algorytmów sortowania, zdecydowanie najkorzystniej wypadło sortowanie typu Heap sort. Po za pojedynczym wynikiem przy losowym typie wprowadzanych liczb wykresy przedstawiające zależność prędkości sortowania do ilości sortowanych liczb wszystkie typy wprowadzanych danych uzyskują w sortowaniu Heap sort liniową notację $O(n)$. Ciężko wskazać najbardziej i najmniej korzystny przypadek użycia tego typu sortowania.

Sortowanie typu insert sort uzyskało największą prędkość sortowania przy korzystnym dla niego typie wprowadzanych danych. W najlepszej sytuacji, gdy dane wprowadzane są w formie już posortowanej w sortowaniu insert sort udało się uzyskać złożoność stałą $O(1)$. Jednak najmniej korzystny dla niej typ danych powodował spadek prędkości sortowania i notację kwadratową $O(n^2)$. W przypadku korzystnych danych ten typ sortowania wydaje się być najlepszy, ale w praktyce rzadko kiedy można spotkać tak optymalne warunki jak, te które wykorzystano w opisywanym eksperymencie.

Sortowanie typu Quick sort uzyskało bardzo dobre wyniki, notację liniową $O(n)$ dla danych wprowadzanych na sposoby losowy, rosnący i malejący, jednak większe problemy sprawił V kształtny typ danych. Quick sort okazało się być drugim pod względem prędkości typem sortowania.

Najwolniejszy typ sortowania – Selection sort mimo delikatnych różnic w poszczególnych pomiarach uzyskał najgorsze wyniki w teście sprawdzającym złożoność obliczeniową algorytmu. Wszystkie typy wprowadzanych danych sortowane były w notacji kwadratowej $O(n^2)$.