

Mikołaj Koszczyk
Zofia Urbaniak

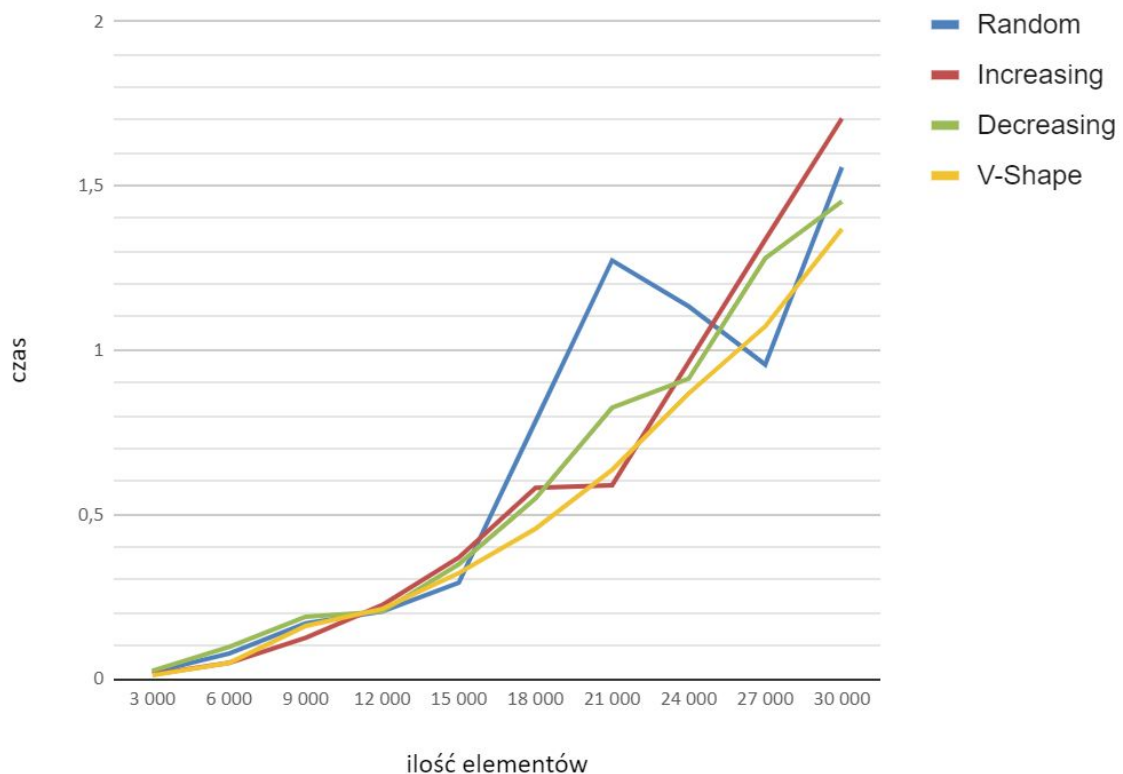
Algorytmy sortowania

Sprawozdanie

1. Selection Sort

	Random	Increasing	Decreasing	V-Shape
3 000	0,019	0,012	0,024	0,010
6 000	0,076	0,048	0,096	0,048
9 000	0,168	0,124	0,188	0,160
12 000	0,204	0,224	0,204	0,212
15 000	0,292	0,368	0,348	0,321
18 000	0,783	0,580	0,548	0,456
21 000	1,272	0,588	0,824	0,635
24 000	1,132	0,963	0,912	0,868
27 000	0,955	1,336	1,279	1,072
30 000	1,556	1,704	1,452	1,368

SelectionSort

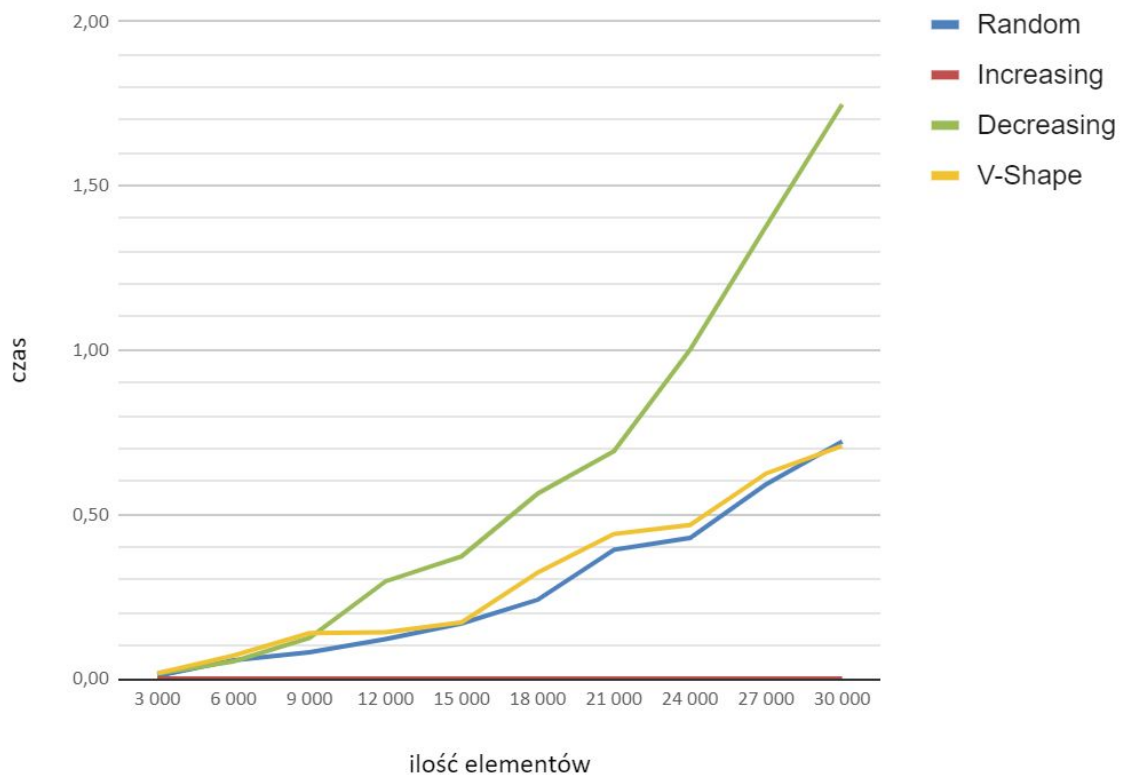


Różnice w szybkości działania algorytmu w zależności od instancji są niewielkie, dlatego bez względu na przypadek, złożoność obliczeniowa zawsze wynosi $O(n^2)$.

2. InsertionSort

	Random	Increasing	Decreasing	V-Shape
3 000	0,008	0,000	0,016	0,016
6 000	0,056	0,000	0,052	0,070
9 000	0,080	0,000	0,124	0,138
12 000	0,120	0,000	0,296	0,141
15 000	0,168	0,000	0,372	0,171
18 000	0,240	0,000	0,563	0,323
21 000	0,392	0,000	0,692	0,440
24 000	0,428	0,000	1,000	0,467
27 000	0,591	0,000	1,376	0,624
30 000	0,721	0,000	1,747	0,708

InsertionSort

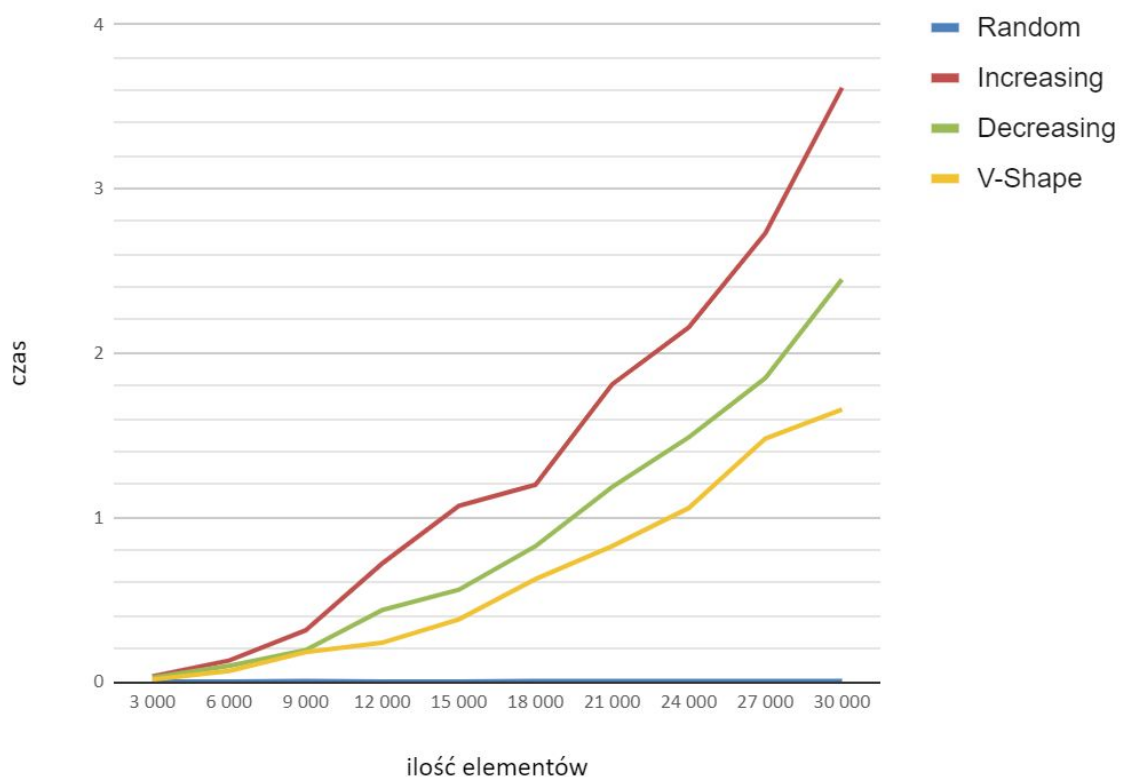


Algorytm działa najszybciej dla danych ułożonych rosnąco, natomiast najwolniejsza, czyli najgorszy przypadek to instancja malejąca, dla której złożoność obliczeniowa wynosi $O(n^2)$.

3. QuickSort

	Random	Increasing	Decreasing	V-Shape
3 000	0,000	0,032	0,024	0,012
6 000	0,000	0,128	0,096	0,064
9 000	0,004	0,312	0,192	0,180
12 000	0,000	0,720	0,436	0,236
15 000	0,000	1,070	0,559	0,379
18 000	0,004	1,198	0,824	0,624
21 000	0,004	1,808	1,184	0,824
24 000	0,004	2,156	1,487	1,056
27 000	0,004	2,729	1,848	1,479
30 000	0,004	3,613	2,446	1,656

QuickSort

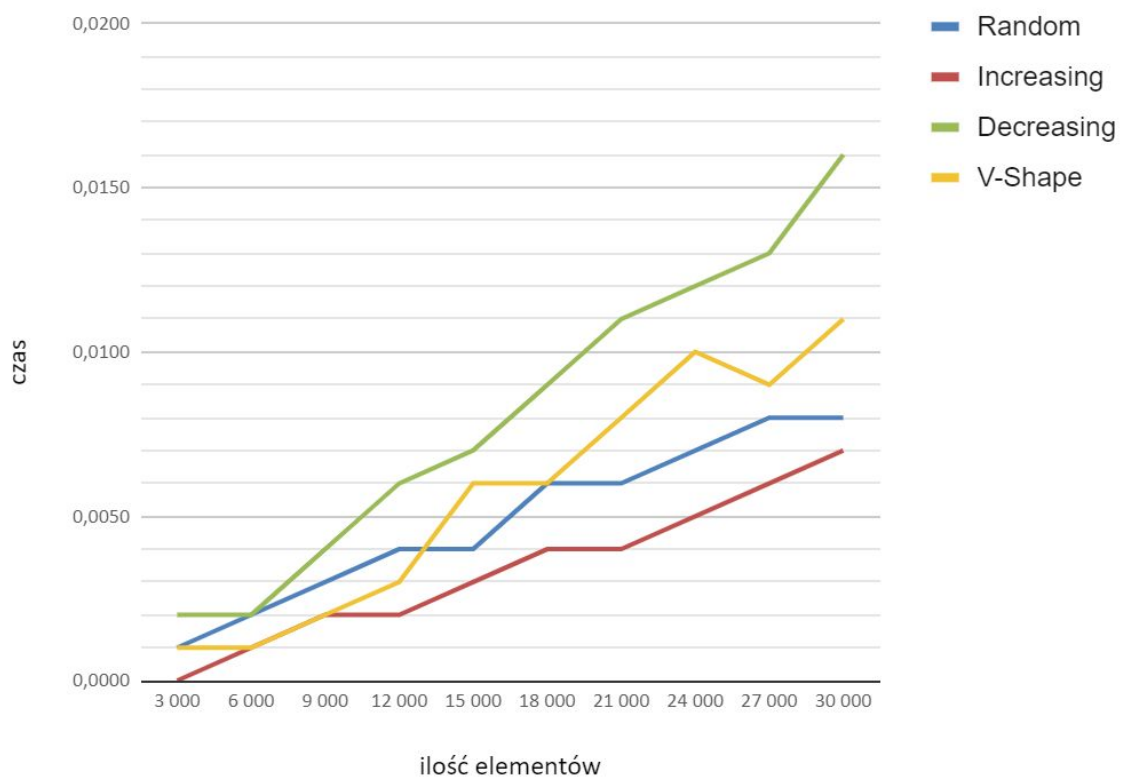


Algorytm Quick Sort działa najszybciej dla wartości losowych, natomiast najwięcej czasu zajmuje posortowanie wartości rosnących i w tym przypadku złożoność obliczeniowa również wynosi $O(n^2)$.

4. HeapSort

	Random	Increasing	Decreasing	V-Shape
3 000	0,001	0,000	0,002	0,001
6 000	0,002	0,001	0,002	0,001
9 000	0,003	0,002	0,004	0,002
12 000	0,004	0,002	0,006	0,003
15 000	0,004	0,003	0,007	0,006
18 000	0,006	0,004	0,009	0,006
21 000	0,006	0,004	0,011	0,008
24 000	0,007	0,005	0,012	0,010
27 000	0,008	0,006	0,013	0,009
30 000	0,008	0,007	0,016	0,011

HeapSort

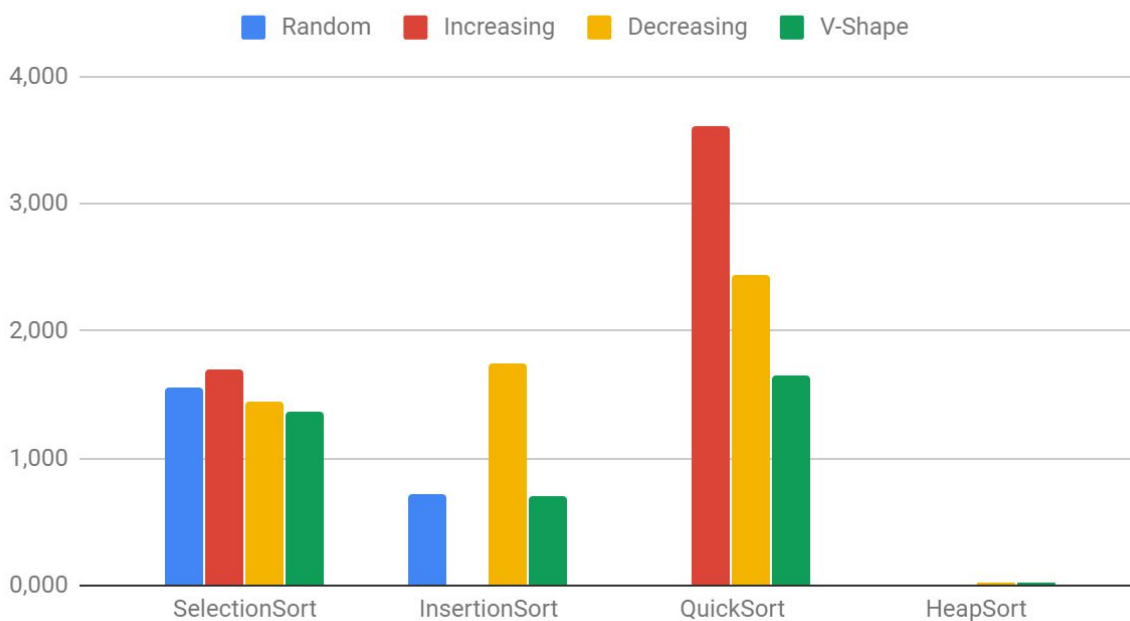


Dla algorytmu Heap Sort najgorszy przypadek zachodzi dla instancji malejącej a złożoność obliczeniowa wynosi wtedy $O(n \log n)$.

5. Porównanie szybkości działania algorytmów dla największych instancji

	SelectionSort	InsertionSort	QuickSort	HeapSort
Random	1,556	0,721	0,004	0,012
Increasing	1,704	0,000	3,613	0,004
Decreasing	1,452	1,747	2,446	0,020
V-Shape	1,368	0,708	1,656	0,016

Porównanie



Jak widać na powyższym wykresie najszybciej działającym algorytmem sortowania jest Heap Sort.