Sprawozdanie

Autor: Sławomir Staniszewski

Nr albumu: 135893

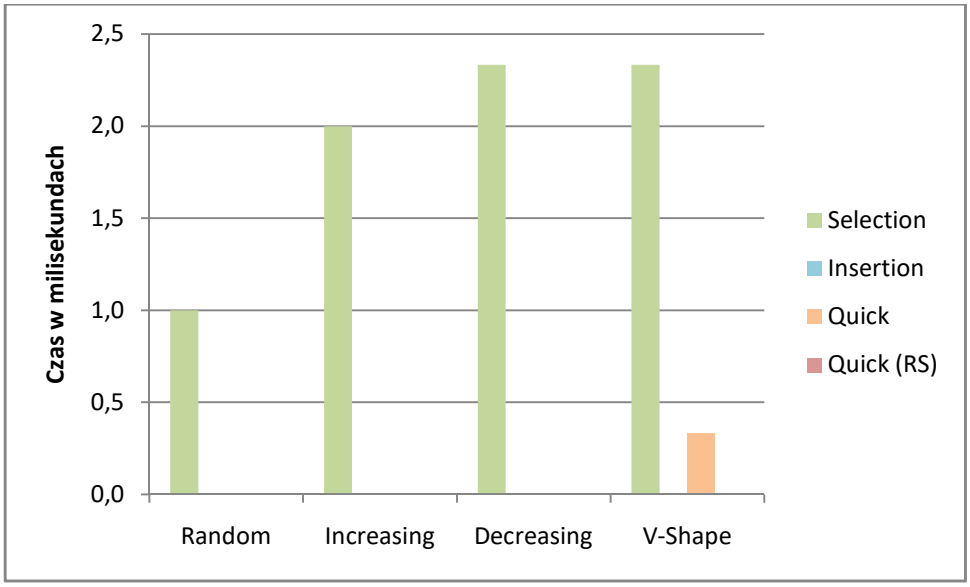
Grupa dziekańska: 3a

Algorytmy i Struktury Danych

Temat: Algorytmy sortowania

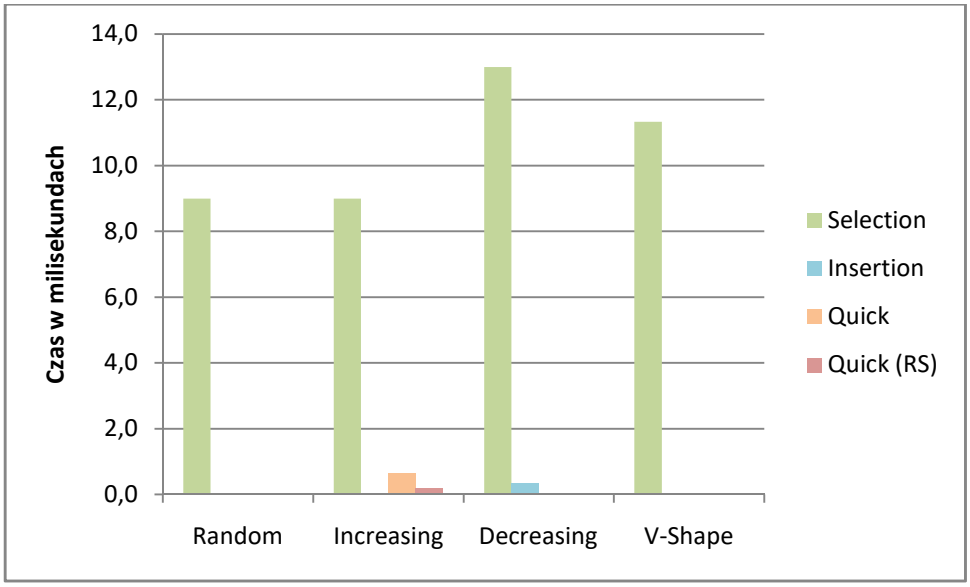
Pomiar czasu sortowania 100 elementów:

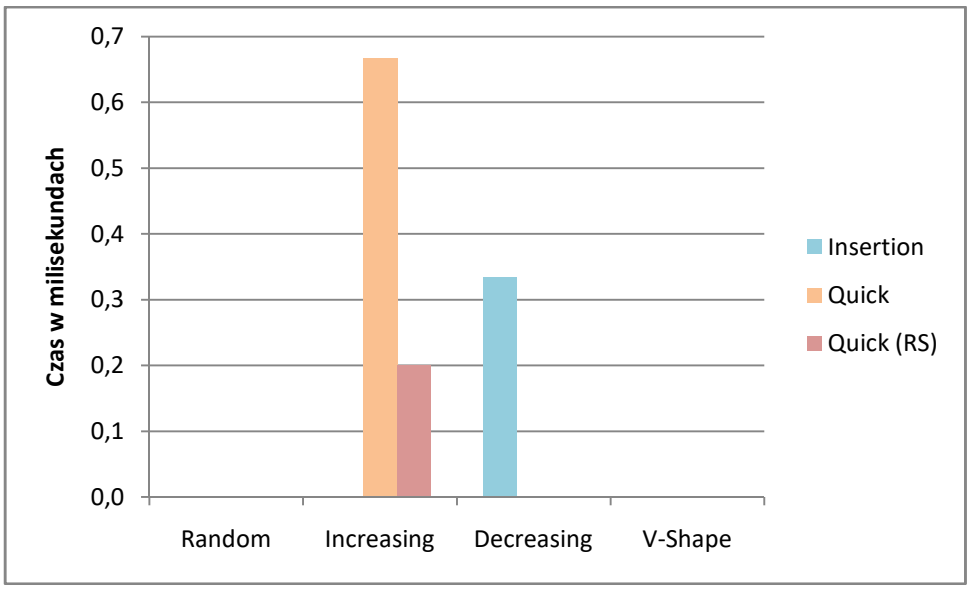
	100 objects			
	Random	Increasing	Decreasing	V-Shape
Selection	1,0	2,0	2,3	2,3
Insertion	0,0	0,0	0,0	0,0
Quick	0,0	0,0	0,0	0,3
Quick (RS)	0,0	0,0	0,0	0,0



Pomiar czasu sortowania 200 elementów:

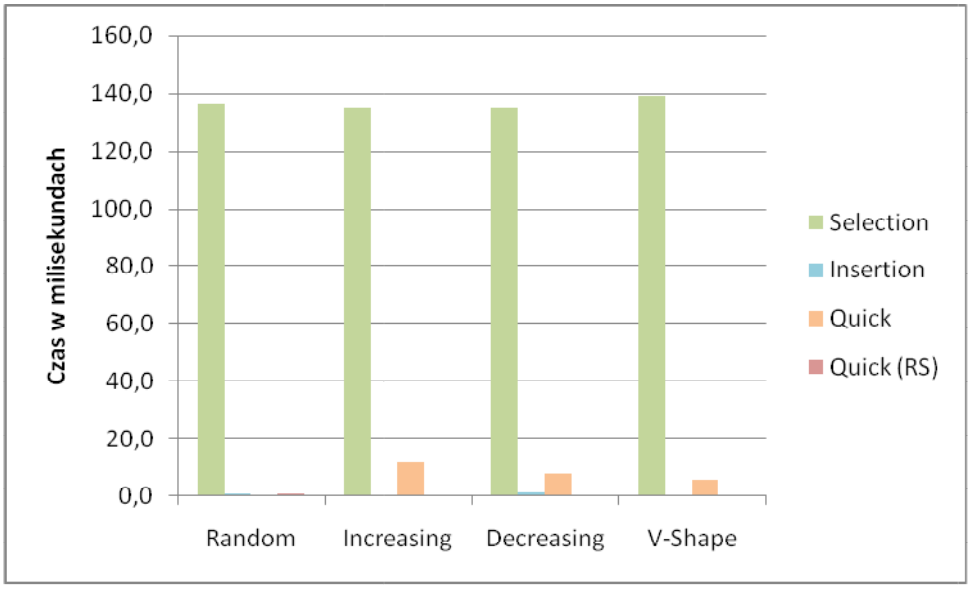
	200 objects			
	Random	Increasing	Decreasing	V-Shape
Selection	9,0	9,0	13,0	11,3
Insertion	0,0	0,0	0,3	0,0
Quick	0,0	0,7	0,0	0,0
Quick (RS)	0,0	0,2	0,0	0,0

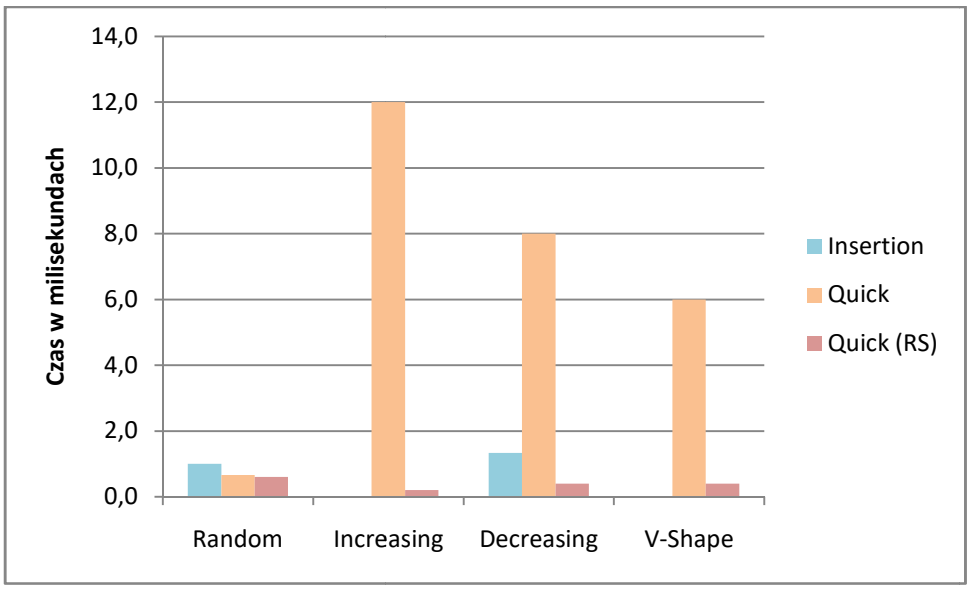




Pomiar czasu sortowania 500 elementów:

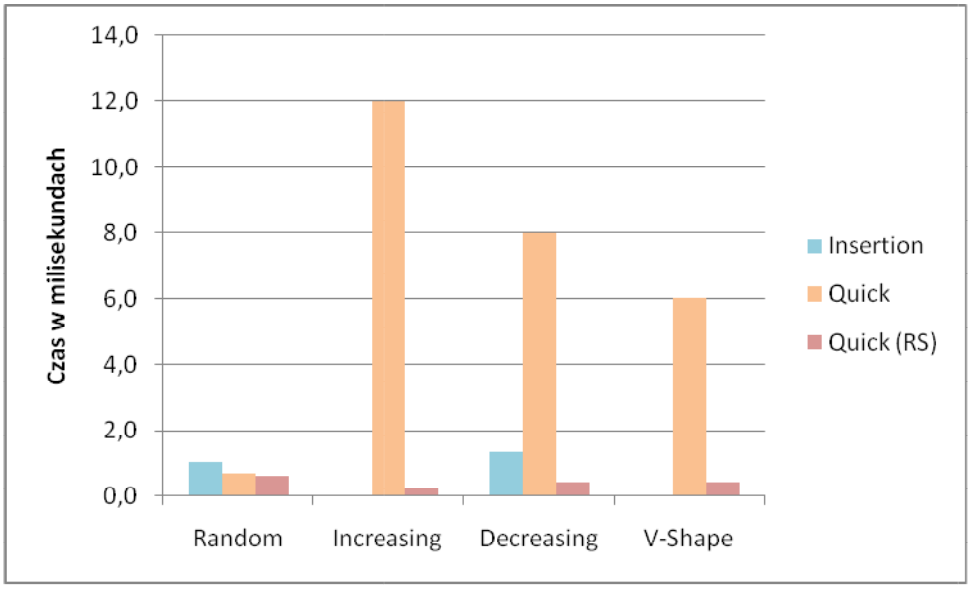
	500 objects			
	Random	Increasing	Decreasing	V-Shape
Selection	136,3	135,0	135,0	139,0
Insertion	0,0	0,0	0,0	0,0
Quick	0,0	3,0	2,0	2,3
Quick (RS)	0,2	0,2	0,2	0,0





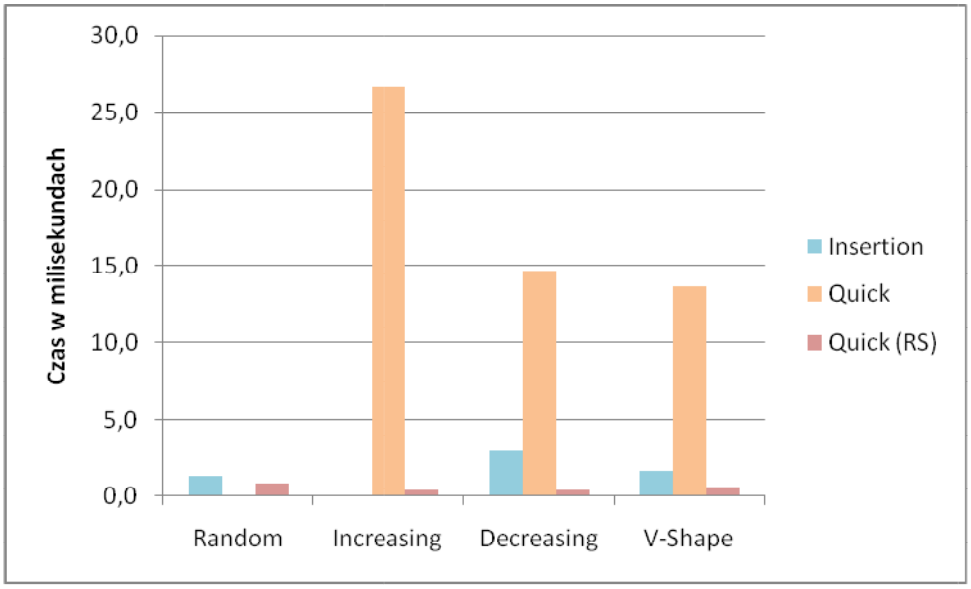
Pomiar czasu sortowania 1000 elementów:

	1000 objects			
	Random	Increasing	Decreasing	V-Shape
Selection	1 078,7	1 070,3	1 068,0	1 067,7
Insertion	1,0	0,0	1,3	0,0
Quick	0,7	12,0	8,0	6,0
Quick (RS)	0,6	0,2	0,4	0,4



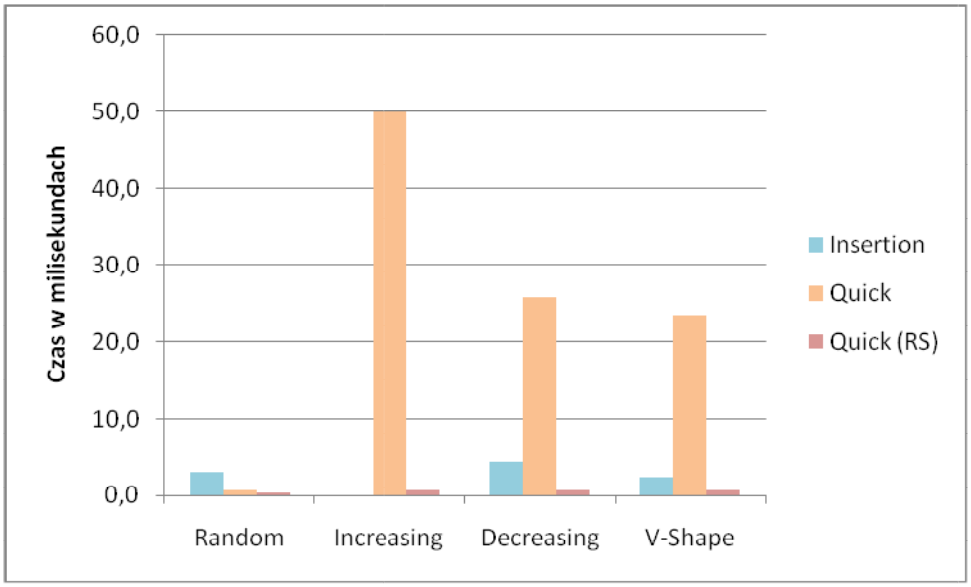
Pomiar czasu sortowania 1500 elementów:

	1500 objects			
	Random	Increasing	Decreasing	V-Shape
Selection	3 618,3	3 612,3	3 602,7	3 593,7
Insertion	1,3	0,0	3,0	1,7
Quick	0,0	26,7	14,7	13,7
Quick (RS)	0,8	0,4	0,4	0,6



Pomiar czasu sortowania 2000 elementów:

	2000 objects			
	Random	Increasing	Decreasing	V-Shape
Selection	8 551,7	8 556,7	8 528,7	8 510,0
Insertion	3,0	0,0	4,3	2,3
Quick	0,7	50,0	25,7	23,3
Quick (RS)	0,4	0,6	0,6	0,6



Podsumowanie

Czas sortowania danych różnej wielkości przez zadane algorytmy (wyrażone w milisekundach):

		Quantity of objects					
		100	200	500	1000	1500	2000
Selection Sort	Random	1,0	9,0	136,3	1 078,7	3 618,3	8 551,7
	Increasing	2,0	9,0	135,0	1 070,3	3 612,3	8 556,7
	Decreasing	2,3	13,0	135,0	1 068,0	3 602,7	8 528,7
	V-Shape	2,3	11,3	139,0	1 067,7	3 593,7	8 510,0
Insertion Sort	Random	0,0	0,0	0,0	1,0	1,3	3,0
	Increasing	0,0	0,0	0,0	0,0	0,0	0,0
	Decreasing	0,0	0,3	0,0	1,3	3,0	4,3
	V-Shape	0,0	0,0	0,0	0,0	1,7	2,3
Quick Sort	Random	0,0	0,0	0,0	0,7	0,0	0,7
	Increasing	0,0	0,7	3,0	12,0	26,7	50,0
	Decreasing	0,0	0,0	2,0	8,0	14,7	25,7
	V-Shape	0,3	0,0	2,3	6,0	13,7	23,3
Quick Sort (Random Splitter)	Random	0,0	0,0	0,2	0,6	0,8	0,4
	Increasing	0,0	0,2	0,2	0,2	0,4	0,6
	Decreasing	0,0	0,0	0,2	0,4	0,4	0,6
	V-Shape	0,0	0,0	0,0	0,4	0,6	0,6

Pomiary wykonano na:
AMD Ryzen 7 2700X, 3,7 GHz, 8/16
16 GB RAM

Złożoność algorytmów:

	Przypadek	
	Najlepszy	*Najgorszy*
Selection	n^2	n^2
Insertion	n	n^2
Quick	$n \log n$	n^2
Quick (RS)	$n \log n$	n^2

Wnioski:
Selection sort – Czas sortowania rośnie bardzo szybko wraz ze wzrostem ilości danych.

Insertion sort – Działa najszybciej, jeśli dane są ułożone rosnąco.

Quick sort – Czas sortowania zależy od tego gdzie nastąpi podział tablicy. Trafianie cały czas w środek pozwala na uzyskanie złożoności logarytmicznej. Jeżeli dzielimy na skrajnych elementach to możemy osiągnąć najgorszą, kwadratową złożoność.

Quick sort (RS) – Dodanie losowej wartości jako elementu rozdzielającego pozwala zmniejszyć szanse na trafianie cały czas w skrajne elementy. Przyspiesza to znacząco szybkość działania algorytmu, co uwidacznia się w powyższych pomiarach.