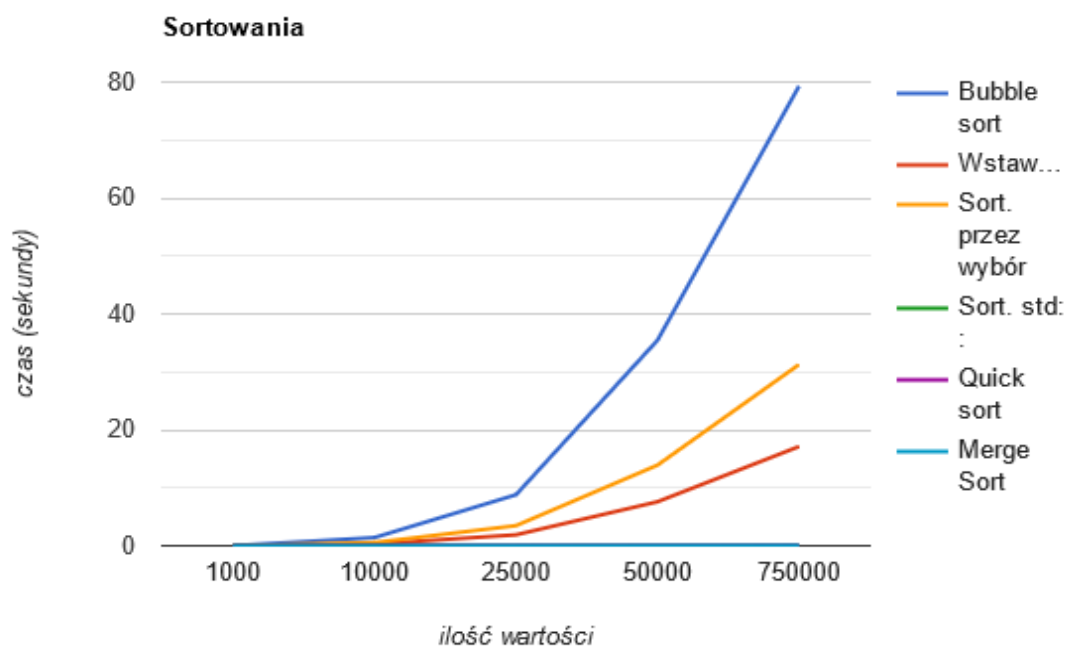


	Próba nr 1					
Ilość wartości:	Bubble sort	Wstawieniowe	Sort. przez wybór	Sort. std.:	Quick sort	Merge Sort
1000	0.0149457	0.0031822	0.0056242	0.0003163	0.0002466	0.0002466
10000	1.41302	0.302658	0.558431	0.0045851	0.0039171	0.0032559
25000	8.80088	1.89159	3.46927	0.0108919	0.0108385	0.0093135
50000	35.4719	7.59904	13.9215	0.0239237	0.0222884	0.0194369
75000	79.327	17.1303	31.2815	0.0352804	0.0330351	0.0301483

	Próba nr 2					
Ilość wartości:	Bubble sort	Wstawieniowe	Sort. przez wybór	Sort. std.:	Quick sort	Merge Sort
1000	0.0143759	0.0032109	0.0058956	0.0003087	0.0002574	0.0002254
10000	1.34496	0.298828	0.549173	0.0040528	0.0036309	0.0031144
25000	9.10198	2.02113	3.77636	0.0110898	0.0101202	0.0088576
50000	36.5172	8.02447	15.1405	0.0251552	0.0214213	0.0185276
75000	82.4463	17.9245	32.9324	0.0360986	0.0312375	0.02928

	Próba nr 3					
Ilość wartości:	Bubble sort	Wstawieniowe	Sort. przez wybór	Sort. std.:	Quick sort	Merge Sort
1000	0.0143925	0.0032949	0.0059396	0.0003184	0.0003038	0.0002941
10000	1.36098	0.297916	0.56094	0.0045936	0.0037434	0.0032068
25000	8.52772	1.87983	mar.56	0.0112564	0.0105842	0.0090959
50000	34.1087	7.50341	13.9424	0.0248914	0.0211132	0.0186097
75000	77.0259	16.9189	31.3833	0.0378789	0.0357559	0.0290124

**WYKRES UŚREDNIONYCH WARTOŚCI:**



Jak widać na wykresie, bubble sort jest algorytmem, który najdłużej radził sobie z sortowaniem. Zaraz po nim, z również nie najlepszymi wynikami znalazły się sortowania przez wybór oraz przez wstawianie. Przechodzenie przez każdą wartość po kolei nie jest dobrym pomysłem. Najszybsze okazały się być sortowanie z biblioteki `std::` oraz sortowania rekurencyjne quick i merge, których na wykresach nawet nie widać, ponieważ z tym, nad czym co bubble sort pracował ponad minutę, one poradziły sobie w mniej niż pół sekundy. Algorytmy, których nie widać na pierwszym wykresie są porównywalnie szybkie. Dzielą one zbiory na mniejsze wartości i wywołują same siebie, co wpływa korzystnie na czas oczekiwania. Dla lepszego zobrazowania warto dodać, że Merge Sort (Sortowania przez scalanie) z 10 milionami wartości poradził sobie w 5 sekund, a `std::sort` w 4,5 sekundy, gdzie bubble sortowi poukładanie 100tys. wartości zajmuje kilka minut. Stworzyłem osobny wykres dla sortowań, które okazały się być szybsze, aby mogły pokazać, na co je stać.

Ilość wartości:	std:: sort	Quick Sort	Merge Sort
1 milion	0.460777	0.535016	0.449368
10 milionów	4.51119	11.009	5.04612
20 milionów	10.3205	35.8821	10.8807

### Sortowania wydajne

