

# Projektowanie algorytmów i metody sztucznej inteligencji

## Projekt 2 - Algorytmy sortowania

Adam Kubiak 249480

14 kwietnia 2020

Wykresy zamieszczone w sprawozdaniu ukazują zależność czasu wykonania algorytmu sortowania  $t(\text{ms})$  od ilości danych w tablicy. Testy zostały przeprowadzone na tablicach: całkowicie losowej, posortowanej w 25%, 50%, 75%, 95%, 99%, 99,7%, i posortowanej w odwrotnej kolejności o rozmiarach: 10000, 50000, 100000, 500000, 1000000. Tablice zostały wypełnione liczbami pseudolosowymi z funkcji `rand()`.

**Quicksort** - Sortowanie to należy do grupy algorytmów, które dzielą problem na podproblemy i wykorzystują rekurencję. Ma on liniowo logarytmiczną złożoność obliczeniową -  $O(n \log(n))$  a w przypadku pesymistycznym kwadratową -  $O(n^2)$ .

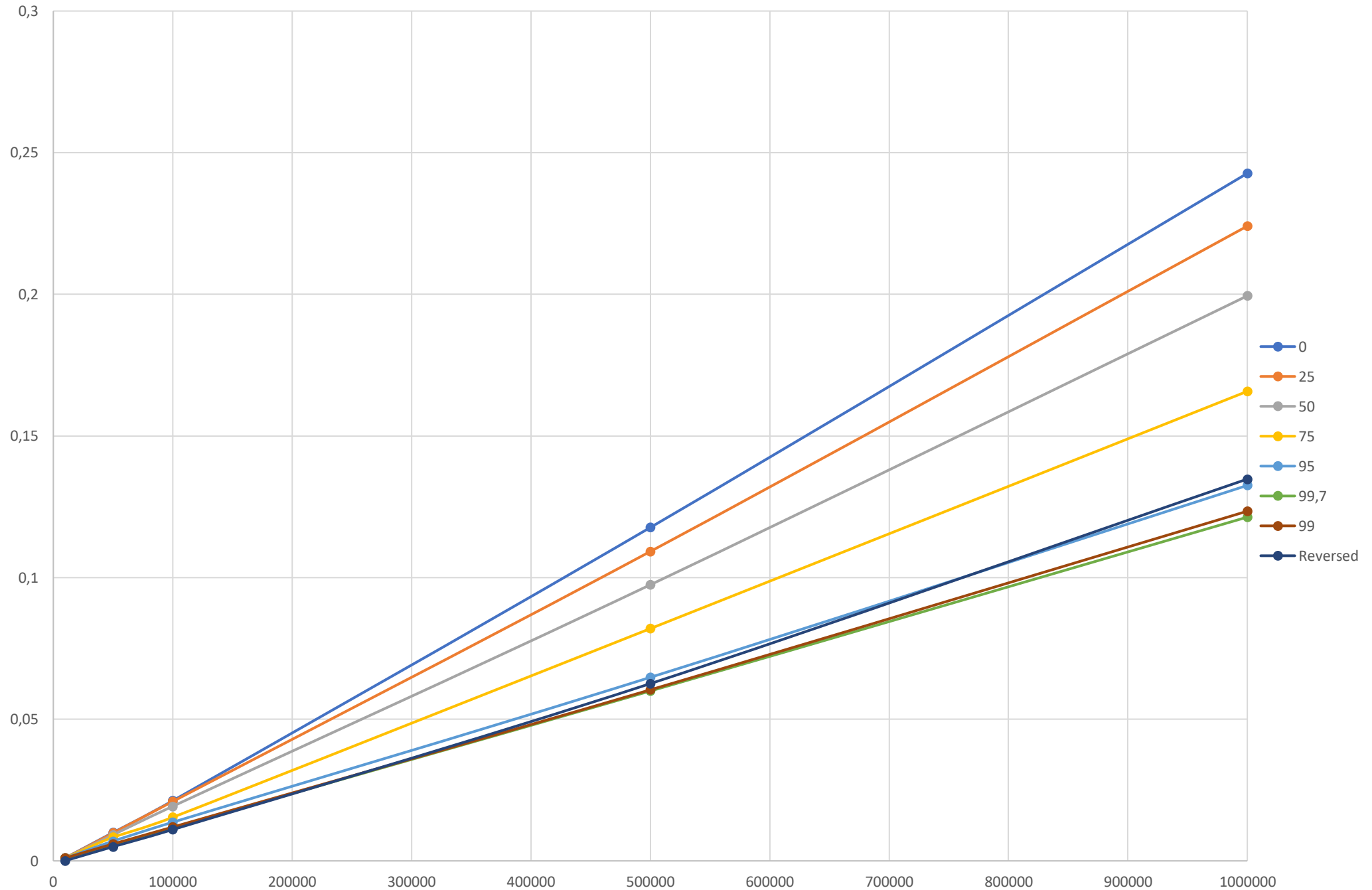
**Mergesort** - Sortowanie to należy do grupy algorytmów, które dzielą problem na podproblemy i wykorzystują rekurencję. Ma on liniowo logarytmiczną złożoność obliczeniową -  $O(n \log(n))$ .

**Introsort** - Odmiana sortowania hybrydowego, w której wyeliminowany został problem złożoności  $O(n^2)$  występującej w najgorszym przypadku algorytmu sortowania szybkiego.

## 1) Quicksort

Percentage	Time [ms]	Number of elements				
		10000	50000	100000	500000	1000000
0	Min	0,001	0,01	0,02	0,116	0,24
	Max	0,002	0,01	0,02	0,124	0,25
	Average	0,00103	0,01	0,02123	0,11769	0,2427
25	Min	0,001	0,009	0,02	0,104	0,213
	Max	0,001	0,010	0,022	0,116	0,238
	Average	0,001	0,00984	0,02096	0,10922	0,22408
50	Min	0,001	0,009	0,018	0,088	0,179
	Max	0,001	0,01	0,021	0,11	0,229
	Average	0,001	0,0092	0,01927	0,09746	0,19945
75	Min	0,001	0,008	0,015	0,078	0,156
	Max	0,001	0,009	0,024	0,095	0,181
	Average	0,001	0,00836	0,01538	0,08199	0,16575
95	Min	0,001	0,006	0,013	0,062	0,13
	Max	0,001	0,007	0,017	0,08	0,141
	Average	0,001	0,00697	0,01365	0,06479	0,13254
99	Min	0,001	0,005	0,011	0,06	0,12
	Max	0,001	0,008	0,013	0,067	0,162
	Average	0,001	0,00598	0,01194	0,06039	0,12344
99,7	Min	0	0,005	0,011	0,059	0,121
	Max	0,001	0,006	0,015	0,066	0,126
	Average	0,00096	0,00571	0,01174	0,05996	0,12135
Reversed	Min	0	0,005	0,011	0,062	0,131
	Max	0,001	0,005	0,014	0,071	0,142
	Average	5,00E-05	0,005	0,01104	0,06257	0,13478

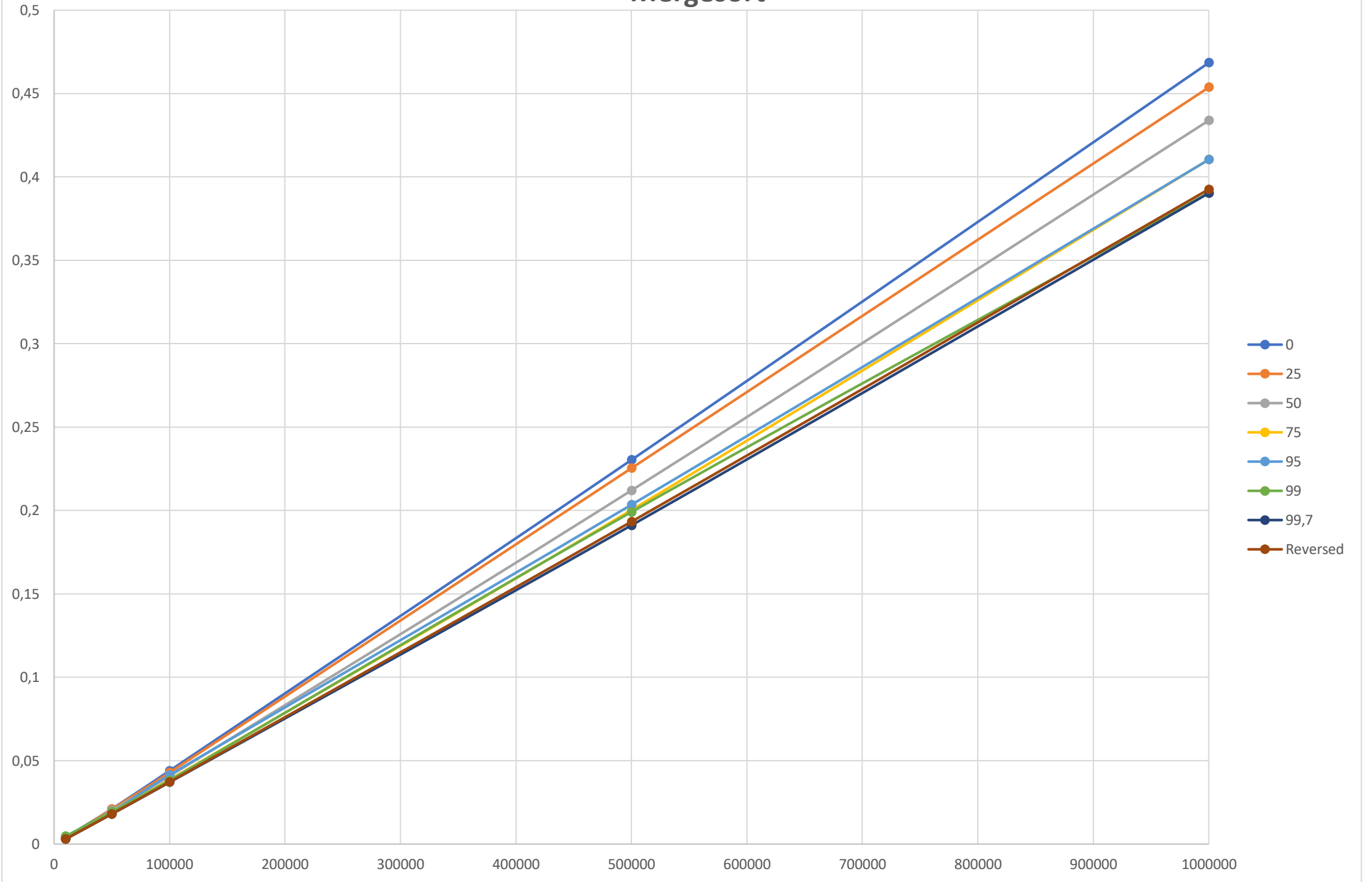
# Quicksort



## 2) Mergesort

Percentage	Time [ms]	Number of elements				
		10000	50000	100000	500000	1000000
0	Min	0,004	0,021	0,043	0,223	0,457
	Max	0,004	0,024	0,046	0,267	0,496
	Average	0,004	0,02105	0,04397	0,23032	0,46845
25	Min	0,003	0,020	0,042	0,217	0,444
	Max	0,004	0,021	0,054	0,291	0,496
	Average	0,004	0,02086	0,0429	0,22536	0,45371
50	Min	0,003	0,02	0,04	0,209	0,423
	Max	0,004	0,025	0,056	0,233	0,486
	Average	0,00323	0,02015	0,04077	0,21205	0,43385
75	Min	0,003	0,019	0,038	0,199	0,405
	Max	0,004	0,023	0,04	0,207	0,456
	Average	0,00302	0,01908	0,03858	0,20028	0,4104
95	Min	0,003	0,018	0,037	0,192	0,395
	Max	0,004	0,024	0,066	0,237	0,503
	Average	0,00304	0,01857	0,0414	0,20345	0,41038
99	Min	0,003	0,018	0,037	0,189	0,388
	Max	0,015	0,031	0,044	0,219	0,402
	Average	0,00467	0,01945	0,03826	0,1991	0,39053
99,7	Min	0,003	0,018	0,037	0,19	0,387
	Max	0,003	0,019	0,039	0,194	0,432
	Average	0,003	0,01802	0,03703	0,19112	0,39026
Reversed	Min	0,003	0,018	0,037	0,192	0,391
	Max	0,003	0,021	0,04	0,195	0,403
	Average	3,00E-03	0,01807	0,03724	0,19331	0,39261

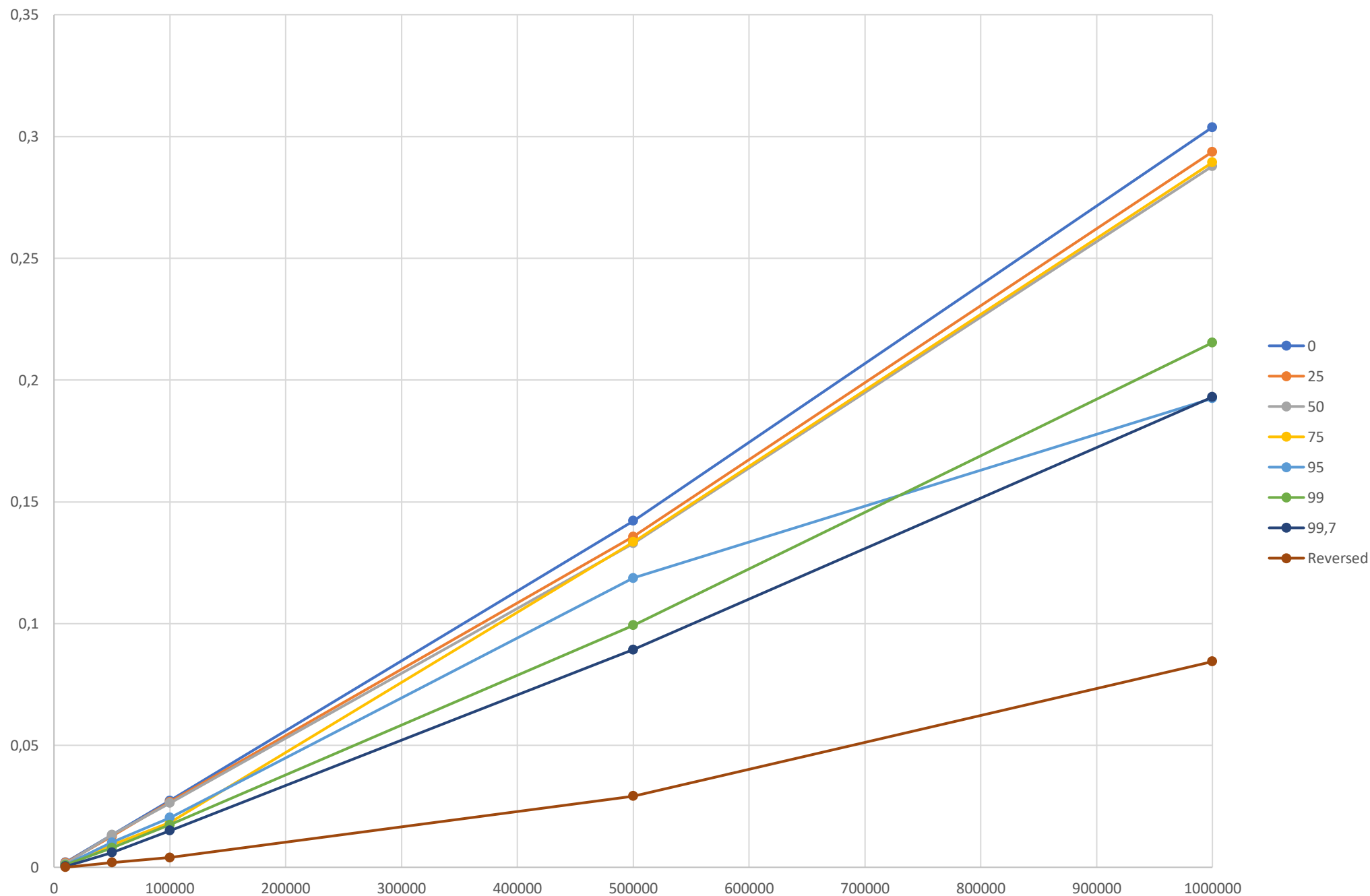
# Mergesort



### 3) Introsort

Percentage	Time [ms]	Number of elements				
		10000	50000	100000	500000	1000000
0	Min	0,002	0,012	0,025	0,131	0,278
	Max	0,002	0,015	0,031	0,173	0,341
	Average	0,002	0,01319	0,02734	0,14221	0,30381
25	Min	0,001	0,011	0,024	0,119	0,251
	Max	0,002	0,016	0,031	0,168	0,376
	Average	0,002	0,01270	0,02684	0,13574	0,29374
50	Min	0,001	0,012	0,023	0,104	0,236
	Max	0,002	0,018	0,033	0,177	0,413
	Average	0,00148	0,01331	0,02637	0,133	0,28783
75	Min	0,001	0,009	0,017	0,107	0,214
	Max	0,001	0,01	0,024	0,17	0,365
	Average	0,001	0,00912	0,01832	0,13353	0,28937
95	Min	0,001	0,009	0,018	0,115	0,171
	Max	0,001	0,013	0,023	0,152	0,238
	Average	0,001	0,01017	0,02039	0,11875	0,1925
99	Min	0	0,007	0,016	0,097	0,213
	Max	0,002	0,01	0,024	0,108	0,231
	Average	0,00101	0,00798	0,01745	0,0994	0,21537
99,7	Min	0	0,006	0,015	0,087	0,188
	Max	0,001	0,007	0,017	0,11	0,208
	Average	0,00032	0,00605	0,01508	0,08931	0,19304
Reversed	Min	0	0,002	0,004	0,028	0,083
	Max	0	0,002	0,005	0,033	0,103
	Average	0,00E+00	0,002	0,00401	0,02921	0,08444

# Introsort



**Wnioski:**

Algorytmy sortowania szybkie i introspektywnego działają w przybliżonych czasach. Dzieje się tak ze względu na podobny styl działania. Dzięki podobnemu działaniu, oraz wiedzy o tych 3 algorytmach zgodnie z założeniami można łatwo zauważyć że sortowania szybkie i introspektywne są szybsze od sortowania przez scalanie. Na wykresie dla sortowania szybkiego widać jak bardzo zmienia się czas w zależności od danych wejściowych. Wyniki pomiarów czasów dla sortowania przez scalanie zgodnie z założeniem tego algorytmu nie wykazują znacznych różnic między maksymalnymi a minimalnymi czasami sortowania(jednak może przez szybkość sprzętu, nie jest to aż tak widoczne w porównaniu z innymi algorytmami, nie miałem możliwości sprawdzenia mojego kodu na innym komputerze). Wynika to z faktu, że dane wejściowe nie mają wpływu na ten algorytm w przeciwieństwie do sortowania szybkiego lub introspektywnego. Czas sortowania przez scalanie wraz z wzrostem ilości elementów tablic, czas rośnie bardzo liniowo i spadek czasu dla bardziej posortowanych tablic jest o wiele mniej zauważalny niż w przypadku innych algorytmów.