

# Projektowanie algorytmów i metody sztucznej inteligencji

Projekt 1  
Algorytmy sortujące  
Paweł Pestka 241513

Prowadzący: dr inż. Łukasz Jeleń  
Termin zajęć: śr. 11:15

## 1. Wstęp

Projekt polegał na implementacji algorytmów sortujących: szybkiego, sortowania przez scalanie, oraz sortowanie introspektywnego. W celu ich porównania konieczne było zmierzenie czasu sortowania tablic o wymiarach 10tys., 50tys., 100tys., 500tys. oraz 1 milion elementów. Dane wejściowe składały się z tablic elementów losowych, posortowanych w 0%, 25%, 50%, 75%, 95%, 99%, 99,7% oraz posortowane w całości lecz w odwrotnej kolejności. Wszystkie z wymienionych algorytmów mają złożoność obliczeniową  $O(n \cdot \log_2 n)$ . Jednak sortowanie szybkie może w przypadku pesymistycznym mieć złożoność na poziomie  $O(n^2)$ . Algorytm sortowania przez scalanie daje nam stabilność nie zwiększając swojej złożoności w przypadku kiedy część tablicy jest już posortowana.

## 2. Sortowanie przez scalanie

Sortowanie przez scalanie stosuje algorytm rekurencyjny. Tablica jest dzielona na podtablice, aż do stworzenia zbiorów jednoelementowych następnie są one scalane w kolejności sortującej i przepisywane z tablicy pomocniczej do podstawowej. Do posortowania tablicy w pamięci wystarczy tablica pomocnicza o wielkości  $n$ . Złożoność obliczeniowa wynosi  $O(n \cdot \log_2 n)$ . Rozpatrując przypadek najgorszy można zauważyć, że zwiększa się jedynie ilość potrzebnych porównań podczas scalania elementów, więc nie powoduje to zmiany złożoności obliczeniowej.

## 3. Sortowanie szybkie

Sortowanie szybkie jest algorytmem rekurencyjnym. Funkcja wywołując się sama wybiera oś służącą do podziału tablicy. Element osiowy jest wysyłany na koniec tablicy, a poszczególne elementy są porównywane tak, by w pierwszej połowie znalazły się elementy mniejsze od elementu osiowego, a w drugiej większe. Następnie element osiowy powraca na środek i funkcja sortowania szybkiego wywoływana jest dla obu podtablic.

Złożoność obliczeniowa jest uzależniona od wyboru elementu podziału jak wspomniałem złożoność obliczeniowa dla średniego przypadku wynosi  $O(n \cdot \log_2 n)$ . W pesymistycznym

przypadku algorytm w każdym kroku może wybrać element największy lub najmniejszy, w tablicy rekurencja będzie musiała wykonać się  $n$  razy, dając złożoność obliczeniową  $O(n^2)$ .

#### 4. Sortowanie introspektywne

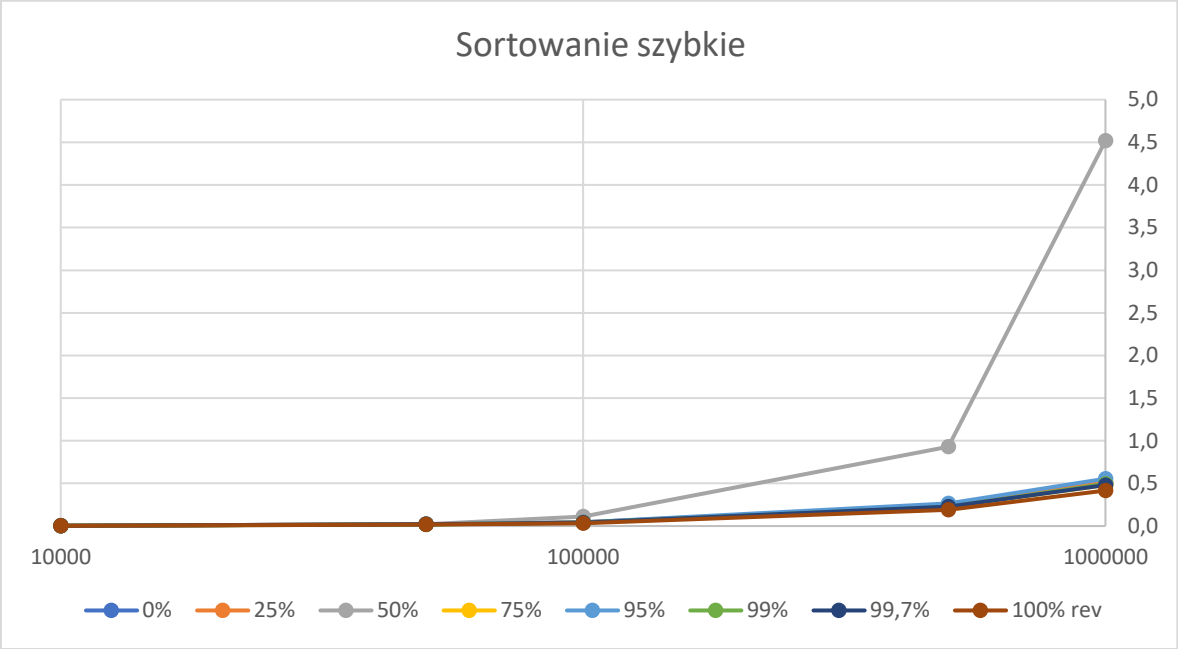
Sortowanie introspektywne łączy w sobie dwa algorytmy sortowania. W podstawowej wersji wykorzystuje sortowanie przez kopcowanie oraz szybkie, jednak można również rozszerzyć algorytmy o sortowanie przez wstawianie. Głównym jej zadaniem jest eliminacja przypadku pesymistycznego w sortowaniu szybkim. W pierwszej kolejności konieczne jest określenie maksymalnej głębokości rekurencji np. na poziomie  $2 \cdot \log_2 n$ . Parametr ten przekazany jako kopia jest wykorzystywany do kontroli sortowania szybkiego, które z każdym wywołaniem zmniejsza wartość kopii, kiedy jest równa 0 następuje zmiana algorytmu a przygotowaną przez Quicksort tablicą zajmie się kopcowanie o złożoności obliczeniowej  $O(n \cdot \log_2 n)$ . Dla dużych tablic jest ono wolniejsze od sortowania szybkiego, jednak jest wywoływane dla dużo mniejszych, dzięki czemu czas całego sortowania jest zbliżony do sortowania szybkiego oraz przyspiesza przypadek najgorszy, gdy Quicksort wybierze za element osiowy wartość największą lub najmniejszą.

#### 5. Pomiary i wykresy

Quicksort		10k	50k	100k	500k	1M
0,00%	min	0,0030	0,0190	0,0390	0,2260	0,4710
	średnie	0,0036	0,0208	0,0427	0,2358	0,5026
	max	0,0060	0,0260	0,0480	0,2530	0,5470
25,00%	min	0,0030	0,0180	0,0400	0,2220	0,4710
	średnie	0,0036	0,0202	0,0438	0,2352	0,5126
	max	0,0050	0,0290	0,0590	0,2560	0,5580
50,00%	min	0,0030	0,0200	0,0430	0,2460	0,5170
	średnie	0,0045	0,0226	0,1129	0,9300	4,5172
	max	0,0080	0,0280	0,8060	7,0760	21,4700
75,00%	min	0,0020	0,0180	0,0390	0,2130	0,4480
	średnie	0,0032	0,0202	0,0414	0,2290	0,4797
	max	0,0040	0,0320	0,0680	0,3160	0,5310
95,00%	min	0,0030	0,0210	0,0430	0,2450	0,5130
	średnie	0,0036	0,0217	0,0456	0,2635	0,5551
	max	0,0050	0,0240	0,0500	0,3070	0,7520
99,00%	min	0,0030	0,0170	0,0380	0,2140	0,4570
	średnie	0,0031	0,0191	0,0393	0,2251	0,4941
	max	0,0050	0,0300	0,0430	0,2450	0,6710
99,70%	min	0,0020	0,0170	0,0370	0,2140	0,4690
	średnie	0,0033	0,0196	0,0400	0,2267	0,4805

	max	0,0080	0,0280	0,0430	0,2740	0,4940
rev	min	0,0020	0,0150	0,0310	0,1830	0,3970
	średnie	0,0030	0,0165	0,0342	0,1919	0,4165
	max	0,0050	0,0200	0,0510	0,2200	0,4680

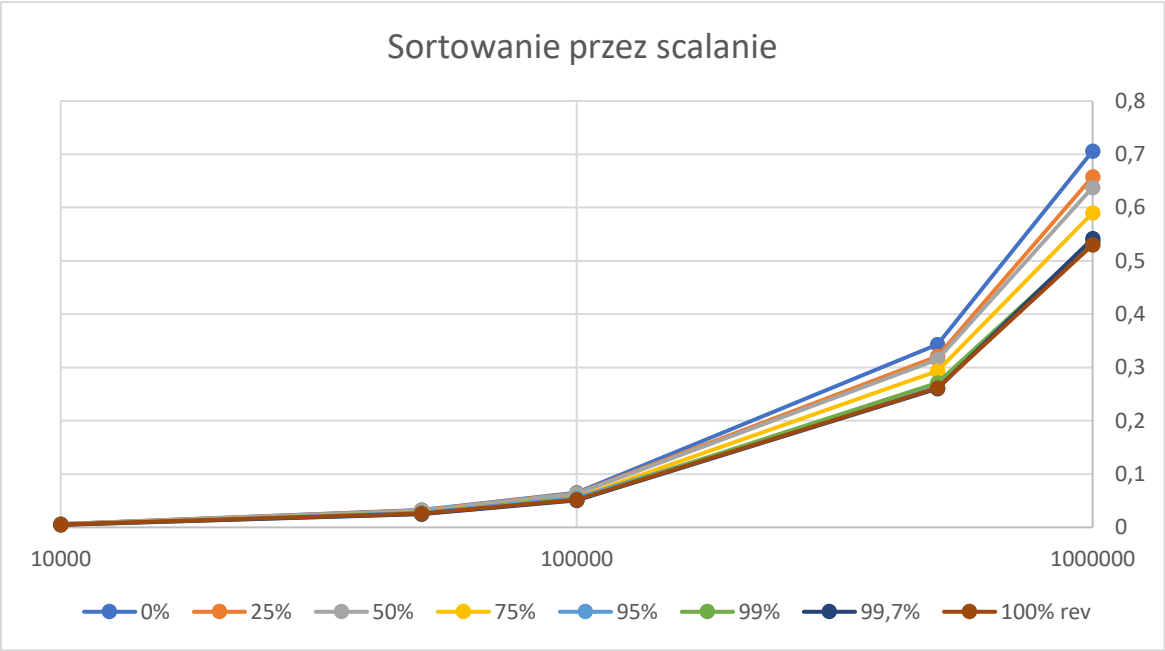
Tabela 1 Pomiary sortowania tablicy sortowaniem szybkim



Scalanie		10k	50k	100k	500k	1M
0,00%	min	0,0050	0,0290	0,0610	0,3300	0,6670
	średnie	0,0062	0,0325	0,0652	0,3429	0,7060
	max	0,0090	0,0390	0,0810	0,3880	0,8270
25,00%	min	0,0050	0,0280	0,0570	0,3100	0,6310
	średnie	0,0061	0,0318	0,0641	0,3211	0,6579
	max	0,0080	0,0420	0,0850	0,3630	0,7150
50,00%	min	0,0040	0,0270	0,0560	0,2990	0,5880
	średnie	0,0060	0,0313	0,0632	0,3151	0,6378
	max	0,0090	0,0440	0,0970	0,3590	0,7430
75,00%	min	0,0040	0,0260	0,0530	0,2810	0,5750
	średnie	0,0056	0,0284	0,0562	0,2934	0,5901
	max	0,0080	0,0330	0,0600	0,3250	0,6080
95,00%	min	0,0040	0,0250	0,0490	0,2490	0,5100
	średnie	0,0054	0,0276	0,0554	0,2641	0,5353
	max	0,0070	0,0370	0,0700	0,2940	0,6100
99,00%	min	0,0040	0,0240	0,0480	0,2540	0,5060
	średnie	0,0050	0,0254	0,0518	0,2712	0,5374
	max	0,0060	0,0290	0,0630	0,3170	0,5830
99,70%	min	0,0040	0,0230	0,0480	0,2510	0,5070
	średnie	0,0048	0,0248	0,0505	0,2601	0,5417
	max	0,0060	0,0300	0,0600	0,2960	0,6490

rev	min	0,0040	0,0250	0,0480	0,2520	0,5120
	średnie	0,0053	0,0260	0,0519	0,2613	0,5301
	max	0,0070	0,0290	0,0590	0,2890	0,5950

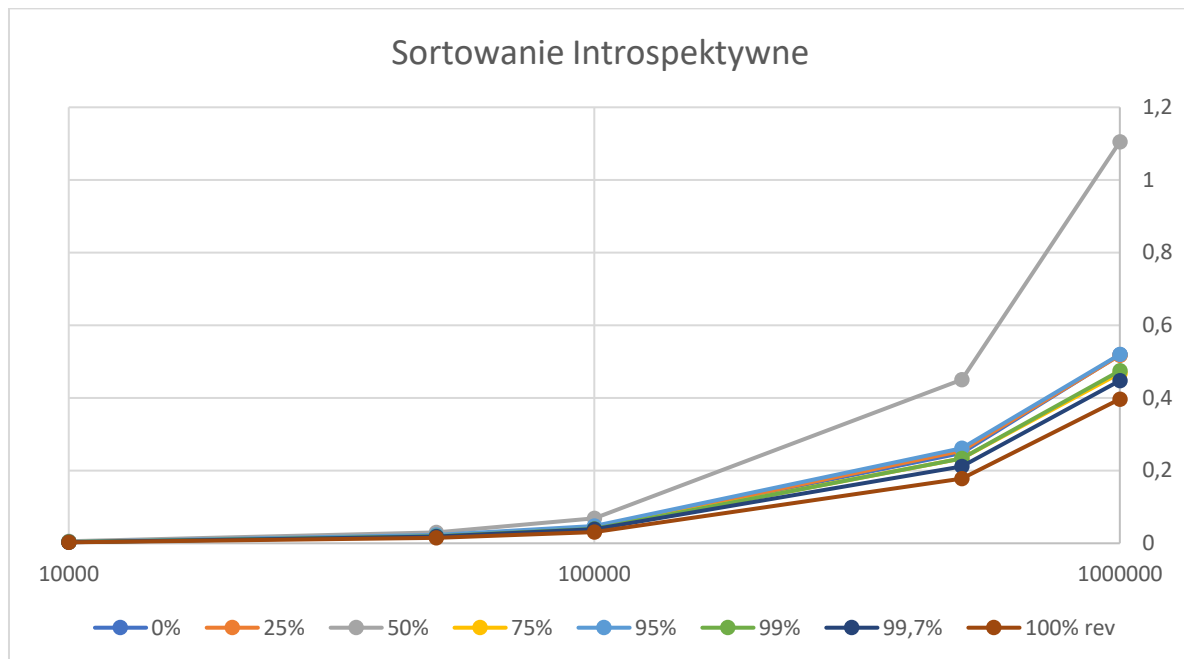
Tabela 2 Pomiary sortowania tablicy sortowaniem przez scalanie



Introspektywne		10k	50k	100k	500k	1M
0,00%	min	0,0030	0,0190	0,0410	0,2340	0,4890
	średnie	0,0041	0,0203	0,0442	0,2484	0,5188
	max	0,0320	0,0220	0,0480	0,2650	0,5950
25,00%	min	0,0030	0,0190	0,0410	0,2280	0,4740
	średnie	0,0035	0,0210	0,0430	0,2545	0,5175
	max	0,0040	0,0240	0,0530	0,2910	0,5640
50,00%	min	0,0030	0,0210	0,0460	0,2480	0,5120
	średnie	0,0057	0,0298	0,0689	0,4505	1,1048
	max	0,0080	0,0570	0,1050	1,0050	1,9440
75,00%	min	0,0030	0,0180	0,0390	0,2240	0,4530
	średnie	0,0034	0,0194	0,0410	0,2336	0,4680
	max	0,0040	0,0210	0,0450	0,2950	0,4900
95,00%	min	0,0030	0,0210	0,0450	0,2560	0,5090
	średnie	0,0038	0,0227	0,0477	0,2615	0,5198
	max	0,0060	0,0270	0,0680	0,2700	0,5340
99,00%	min	0,0030	0,0180	0,0380	0,2260	0,4650
	średnie	0,0033	0,0196	0,0399	0,2332	0,4747
	max	0,0040	0,0230	0,0460	0,2560	0,5390
99,70%	min	0,0020	0,0160	0,0350	0,2080	0,4420
	średnie	0,0029	0,0181	0,0387	0,2117	0,4476
	max	0,0050	0,0210	0,0480	0,2230	0,4570
rev	min	0,0020	0,0140	0,0290	0,1710	0,3880

	średnie	0,0026	0,0152	0,0312	0,1777	0,3969
	max	0,0040	0,0200	0,0440	0,2380	0,4280

Tabela 3 Pomiary sortowania tablicy sortowaniem Introspektywnym



## 6. Wnioski

Opracowane algorytmy pozwalają bardzo szybko posegregować nawet kilkanaście milionowo elementowych tablic każda. Sortowanie szybkie trafiło na pesymistyczny przypadek w momencie sortowania tablic wcześniej posortowanych w 50%. Na szczęście sortowaniu Introspektywnemu udało się ograniczyć średni czas sortowania w tym przypadku. W pozostałych pomiarach Quicksort najlepiej poradził sobie ze swoim zadaniem. Sortowanie przez scalanie okazało się najwolniejsze. Wszystkie algorytmy najszybciej poradziły sobie z tablicą wstępnie nieposortowaną.

## 7. Bibliografia

- [www.wikipedia.pl](http://www.wikipedia.pl)
- „Wprowadzenie do algorytmów” Thomas H. Cormen, Charles E. Leiserson.