

Politechnika Wrocławska
Wydział Elektroniki
Sprawozdanie z projektu

Projektowanie Algorytmów i Metody Sztucznej Inteligencji

Algorytmy Sortowania

Prowadząca: Mgr inż. Marta Emirsajłow

Autor: Patrick Rossol 249470

Grupa: Piątek 13:15-15:00

1. Wprowadzenie

Zadaniem projektu pierwszego jest zapoznanie się z różnymi implementacjami algorytmów sortowania oraz zapoznanie się z Big-O notation. Zaimplementowanymi algorytmami sortowania są sortowanie szybkie, sortowanie przez scalanie i sortowanie introspektywne. Są to algorytmy o złożoności $O(n \log n)$ lecz dla sortowania szybkiego najgorszy przypadek ma $O(n^2)$.

2. Opis Algorytmów

2.1 Quicksort - Algorytm sortowania szybkiego działa na zasadzie dzielenia tablicy i sortowania ich elementów. Wybierany jest element dzielący dla którego tworzone są nowe tablice, jedna z większymi elementami, a druga z mniejszymi niż element dzielący. Następnie dla tych tablic wywoływany jest rekurencyjnie ten sam algorytm. Złożoność obliczeniowa algorytmu zależy od elementu dzielącego. W średnim rozrachunku algorytm ma złożoność $O(n \log n)$ dlatego można uznać go za wydajny, lecz jeśli element dzielący jest największy lub najmniejszy lub blisko ich wartości to złożoność rośnie do $O(n^2)$.

Złożoność obliczeniowa:

Średni przypadek: $O(n \log n)$

Najgorszy przypadek: $O(n^2)$

2.2 Mergesort - algorytm sortowania przez scalanie polega na podobnej zasadzie co algorytm sortowania szybkiego tzn. dziel i zwyciężaj. Różnicą jest to że ten algorytm jest algorytmem stabilnym i jego złożoność obliczeniowa wynosi $O(n \log n)$, bez najgorszego przypadku. Zasada działania algorytmu polega na dzieleniu danych na 2 mniejsze części i następnie podzieleniu ich na kolejne mniejsze części do momentu aż pozostaje jeden element. Gdy wszystkie elementy są już podzielone na pojedyncze dane należy je scalać i sortować mniejsze zestawy danych na większe aż do uzyskania całego ciągu.

Złożoność obliczeniowa:

Średni przypadek: $O(n \log n)$

Najgorszy przypadek: $O(n \log n)$

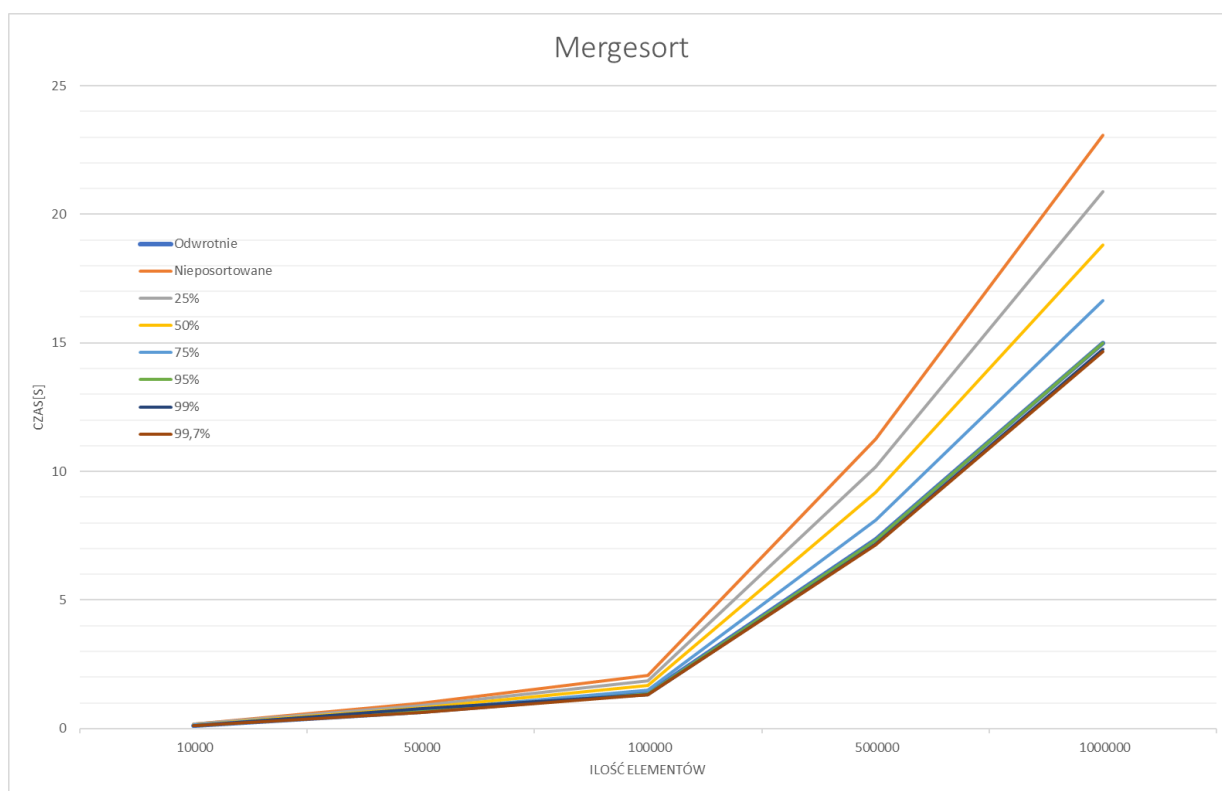
2.3 Introsort - to algorytm hybrydowy tzn. że posiada elementy innych algorytmów sortowania. Jest to połączenie sortowania przez kopcowanie i sortowania szybkiego z wykorzystaniem sortowania przez wstawianie dla małych zestawów danych. Działa on podobnie co algorytm sortowania szybkiego na początku wybierając element dzielący i wywołuje się ten sam algorytm dla podzielonych zestawów. Badana w tym algorytmie jest głębokość rekurencji i kiedy osiągniemy maksymalny poziom obliczany wzorem $2 \log(n)$, gdzie n jest długością ciągu, pozostałą tablicę sortuje się przez kopcowanie. Algorytm sortowania przez wstawianie jest stosowany w przypadku gdy długość zestawu danych będzie krótsza niż dany próg. W ten sposób algorytm rozpatruje najgorszy przypadek sortowania szybkiego $O(n^2)$ i powoduje że będzie miał złożoność bliską $O(n \log n)$.

Złożoność obliczeniowa:

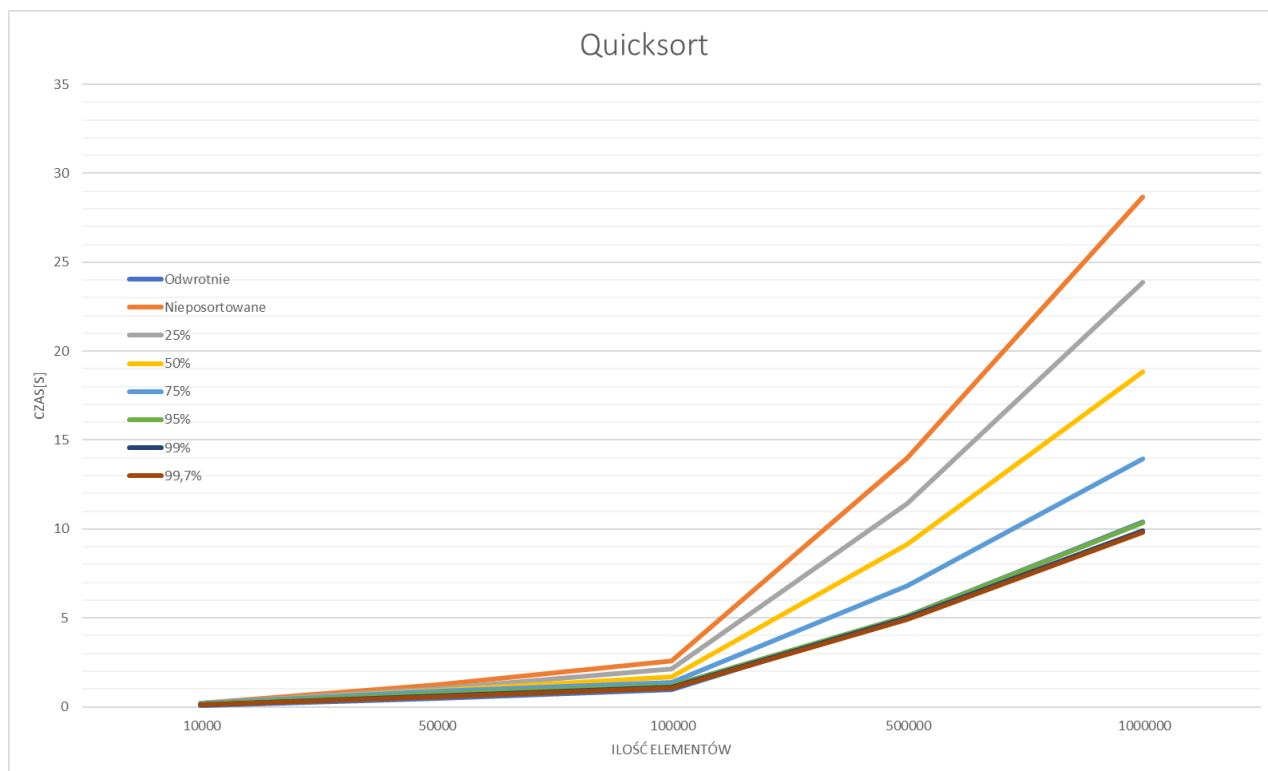
Średni przypadek: $O(n \log n)$

Najgorszy przypadek: $O(n \log n)$

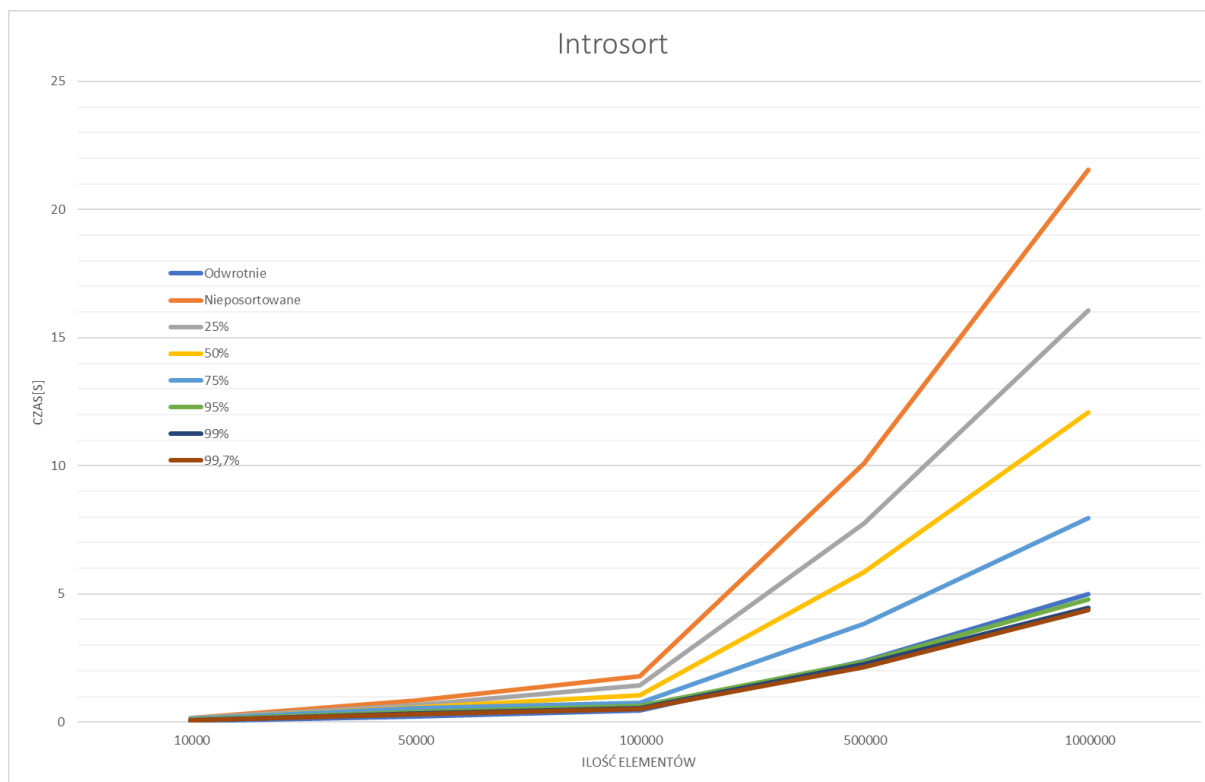
3. Tabele i wykresy



Dla jednej tablicy		10000	50000	100000	500000	1000000
Odwrotnie	śrd	1,46548	7,83341	16,0783	85,2017	174,97
	min	1,39141	7,2234	15,1244	84,412	169,12
	max	1,55236	7,9213	16,2578	86,235	175,214
Nieposrtowane	śrd	1,34761	7,26437	14,8162	79,3063	162,873
	min	1,3014	7,2141	14,5321	77,985	161,42
	max	1,7213	8,1231	15,1231	81,432	163,42
25%	śrd	1,22858	6,71676	13,8413	73,0373	150,767
	min	1,2116	6,2342	13,2673	72,121	149,512
	max	1,32441	7,1232	14,2131	74,2142	151,124
50%	śrd	1,11842	6,13076	12,4993	67,0677	138,733
	min	0,8743	5,1256	11,1241	66,1532	136,121
	max	1,2142	6,4241	12,792	68,124	139,1532
75%	śrd	1,02272	5,6384	11,6454	63,1535	134,148
	min	0,9144	5,1124	10,8745	62,567	133,674
	max	1,1244	6,0124	12,442	63,6723	135,573
95%	śrd	1,00715	5,49068	11,4554	61,3295	127,179
	min	0,9153	5,1042	10,4211	60,2124	126,214
	max	1,5672	5,7823	12,1452	62,1245	128,552
99%	śrd	1,00471	5,40144	11,2967	60,8584	125,985
	min	0,8997	5,3213	10,1221	59,3214	124,124
	max	1,1241	5,8992	11,9621	62,1452	126,245
99,7%	śrd	1,0086	5,46186	11,512	62,4407	128,57
	min	0,9521	4,9561	11,003	61,795	127,12
	max	1,1241	5,9329	12,124	63,425	129,53



Dla jednej tablicy		10000	50000	100000	500000	1000000
Odwrótnie	śrd	1,85236	10,2659	18,5236	118,406	245,338
	min	1,63241	9,2234	17,1314	116,412	245,338
	max	1,85236	11,3213	18,8754	118,406	245,338
Nieposrtowane	śrd	1,56199	8,61289	17,8623	97,442	202,554
	min	1,42312	7,21341	15,21231	96,213	201,211
	max	1,7213	9,1231	18,1231	99,432	214,12
25%	śrd	1,25881	6,9009	14,3067	77,7054	160,925
	min	1,2016	6,2342	13,2673	75,934	157,543
	max	1,34251	7,3145	15,2131	79,1247	165,124
50%	śrd	0,969415	5,22692	10,8197	58,4064	121,329
	min	0,8743	4,6123	9,5631	56,1522	119,312
	max	1,1213	6,5121	12,792	60,632	124,734
75%	śrd	0,76025	3,99499	8,20506	44,0219	89,9164
	min	0,75222	3,5123	7,89641	42,1333	87,2376
	max	0,8213	4,5212	8,9642	45,7641	92,1245
95%	śrd	0,733391	3,76295	7,74175	41,2422	84,317
	min	0,69213	3,41512	6,6342	39,1112	82,4121
	max	0,761245	4,2412	8,5152	43,1225	86,1267
99%	śrd	0,746487	3,73995	7,67016	40,9568	83,2761
	min	0,68113	3,2412	6,5123	40,1215	82,1255
	max	0,77123	4,1267	7,9654	42,163	84,634
99,7%	śrd	0,828822	4,13522	8,52284	45,4919	91,2004
	min	0,79549	3,97864	8,3215	44,1245	89,4526
	max	85,23125	4,2424	8,7412	46,6322	93,6231



Dla jednej tablicy		10000	50000	100000	500000	1000000
Odwrotnie	śrd	1,16424	6,56569	13,7687	78,9835	166,324
	min	1,6234	6,2134	12,9672	77,534	164,122
	max	1,85236	7,4211	14,2314	79,4124	168,124
Nieposrtowane	śrd	0,887373	5,14425	10,8917	62,5907	131,843
	min	0,8624	5,0124	10,1256	61,3456	130,524
	max	0,93221	5,3455	11,6342	63,4412	132,523
25%	śrd	0,668264	3,79828	8,14991	46,406	98,0825
	min	0,65342	3,70123	7,5634	45,502	96,232
	max	0,67455	3,9421	9,0031	47,235	99,412
50%	śrd	0,460631	2,58879	5,5284	30,6971	64,989
	min	0,4402	2,4954	5,1241	29,5621	63,211
	max	0,4712	2,7894	5,9785	31,5311	65,214
75%	śrd	0,31618	1,68083	3,56278	19,3111	40,7774
	min	0,2941	1,5932	3,4215	19,1123	40,2411
	max	0,3314	1,7234	3,6542	19,5633	40,9344
95%	śrd	0,286644	1,51909	3,23829	17,3897	36,666
	min	0,21244	1,4422	2,9531	16,9241	36,101
	max	0,34511	1,6363	3,5642	17,9421	37,231
99%	śrd	0,286644	1,51909	3,23829	17,3897	36,666
	min	0,2402	1,1244	3,1132	16,7651	36,154
	max	0,3245	1,6672	3,6421	17,9421	37,012
99,7%	śrd	0,355708	1,8703	3,92484	20,8665	43,4153
	min	0,3341	1,7643	3,7623	20,1011	42,8962
	max	0,37455	1,9871	4,1242	21,4252	43,9865

4. Wnioski

Przy porównaniu wszystkich algorytmów algorytm szybkiego sortowania dla wartości losowych wypada najłabiej. Jest to związane z wyborem wartości dzielącej. W algorytmie była to wartość środkowa bez kontroli tego, czy jest to wartość największa lub najmniejsza. Ewidentnie widać, że czas sortowania jest dłuższy niż w pozostałych 2 metodach. Jednak porównując algorytm sortowania przez scalanie i sortowanie szybkie przy posortowanych tablicach, widać, że sortowanie przez scalanie ma dłuższy czas niż sortowanie szybkie. Pokazuje to, że dane wejściowe nie mają znaczenia dla tego algorytmu inaczej niż w sortowaniu szybkim i introspektywnym. W sortowaniu Introspektywnym widać najlepsze wyniki. Dla parametru głębokości rekurencji równym $2 \cdot \log(n)$, gdzie n jest długością ciągu oraz dla sortowania przez wstawianie dla tablic mniejszych niż 16 elementów algorytm miał najlepszą wydajność. Zmiana wielkości tablicy na 9 lub 25 elementów skutkowałą zmianą efektywności na gorszą. Również zamiana algorytmu przez wstawianie na algorytm szybkiego sortowania spowodował, że czas nieznacznie się polepszył lub nawet pogorszył. Pokazuje to, że algorytm szybkiego sortowania dla małych tablic nie jest najlepszym algorytmem. Do pomiaru czasu w programie użyta została biblioteka `chrono`. Program po uruchomieniu zaczyna sortować tablice wygenerowane w zależności od typu losowania i od wielkości. Wszystkie wyniki zapisuje w pliku *.csv. Dodatkowo generuje pliki *.txt dla każdej 100 tablic z opisem w nazwie pliku jakie to było sortowanie i jaka tablica. Po wykonaniu jednego sortowania tablice są zwalniane z pamięci w pętli. Po zakończeniu sortowania program przechodzi do następnej linii i czeka na znak końca linii. Po tym program kończy działanie

5. Bibliografia

- <https://en.wikipedia.org/wiki/Quicksort>
- <https://en.wikipedia.org/wiki/Introsort>
- <https://www.geeksforgeeks.org/introsort-or-introspective-sort/>
- <https://www.geeksforgeeks.org/merge-sort/>
- <https://www.geeksforgeeks.org/heap-sort/>
- <https://www.geeksforgeeks.org/quick-sort/>