Małgorzata Pietras
Automatyka i Robotyka
235794

dr inż. Łukasz Jeleń
środa 7:30

Projektowanie Algorytmów i Metody Sztucznej Inteligencji

Projekt 1
data oddania: 03.04.2019

1 Wstęp

Projekt pierwszy polegał na zaimplementowaniu i przeprowadzaniu badań dotyczących złożoności obliczeniowej wybranych algorytmów sortujących. W tym sprawozdaniu omówione są wyniki dla sortowania przez scalanie, szybkiego oraz introspektywnego. Testy wykonane były na tablicach o rozmiarach: 10000, 50000, 100000, 500000 i 1000000. Stopnie posortowania to: $0\%, 25\%, 50\%, 75\%, 95\%, 99\%, 99.7\%$ i tablica posortowana odwrotnie.

2 Opis badanych algorytmów

Sortowanie przez scalanie to algorytm rekurencyjny, w którym tablica dzielona jest aż do powstania tablic jednoelementowych. Jego nazwa związana jest z następującym później procesem scalania tablic. Jest to algorytm o złożoności obliczeniowej $O(n \cdot log_2 n)$, co wynika z podziału każdej tablicy na pół, porównywania ich oraz scalania. Dla tego algorytmu przypadek najgorszy to po prostu przypadek tablicy o największym rozmiarze, lecz nie zmienia się wówczas złożoność obliczeniowa. Sortowanie to jest stabilne, podczas sortowania występujące po sobie jednakowe elementy nie są zamienione miejscami.

Sortowanie szybkie to algorytm rekurencyjny, w którym w każdym kroku sortowania zostaje wybrany element służący do podziału tablicy. To od niego zależy jaką złożoność obliczeniową będzie miał algorytm. Następnie porównywane są do niego wszystkie sortowane elementy. Wartości mniejsze przekazywane są do jednej tablicy, a większe do drugiej. Algorytm działa aż do uzyskania tablic jednoelementowych. Jeżeli rozpatrywana jest opcja optymistyczna i element rozdzielający został określony poprawnie, wówczas złożoność obliczeniowa wynosi $O(n \cdot log_2 n)$ (jest to też złożoność uśredniona). W przypadku pesymistycznym złożoność ta może wynosić nawet $O(n^2)$.

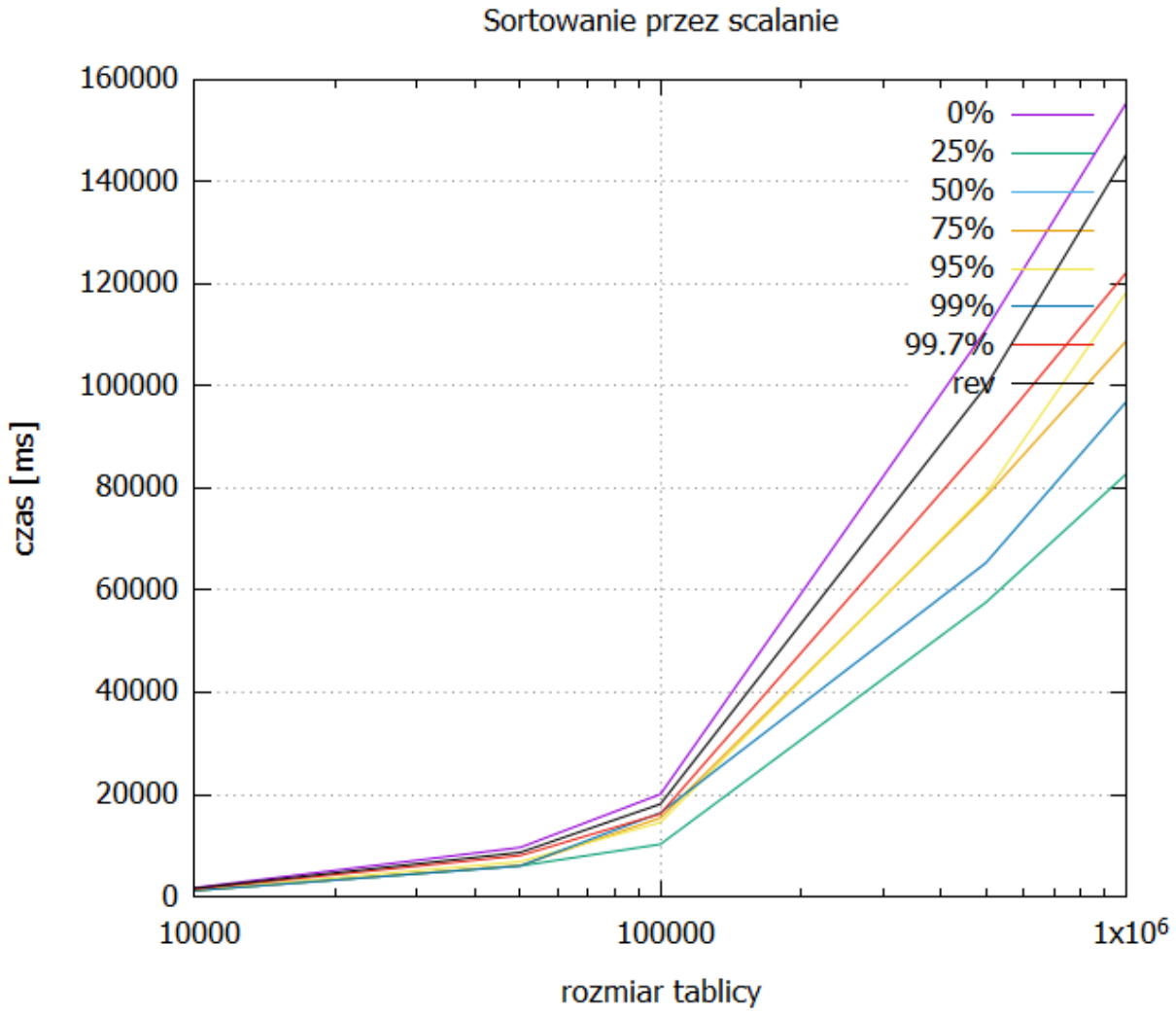
Sortowanie introspektywne jest sortowaniem które łączy ze sobą sortowanie szybkie oraz przez kopcowanie. Stosuje się go w celu uniknięci wariantu pesymistycznego w sortowaniu szybkim, kiedy złożoność obliczeniowa wynosi $O(n^2)$. W tym algorytmie należy obliczyć maksymalną głębokość rekurencji oznaczaną jako M. W wykorzystanych przez mnie źródłach była ona ustalona jako $2 \cdot log_2 n$. Podczas kolejnych, rekurencyjnych wywołań programu (w którym tablice są dzielone, podobieństwo do sortowania szybkiego) wartość M jest zmniejszana o 1. Gdy maksymalna głębokość rekurencji będzie równa zero stosuje się sortowanie przez kopcowanie. Złożoność obliczeniowa połączonych algorytmów to $O(n \cdot log_2 n)$.

3 Przebieg eksperymentu i uzyskane wyniki

Przeprowadzone obliczenia polegały na badaniu czasu działania poszczególnych algorytmów sortujących na stu tablicach. Po wprowadzaniu rozmiaru tablicy oraz sposobu jej uporządkowania pomiary czasu wyrażone w milisekundach zostały zapisane do plików w celu dalszej analizy. Wyniki w postaci wykresów oraz tablic zostały pokazane poniżej.

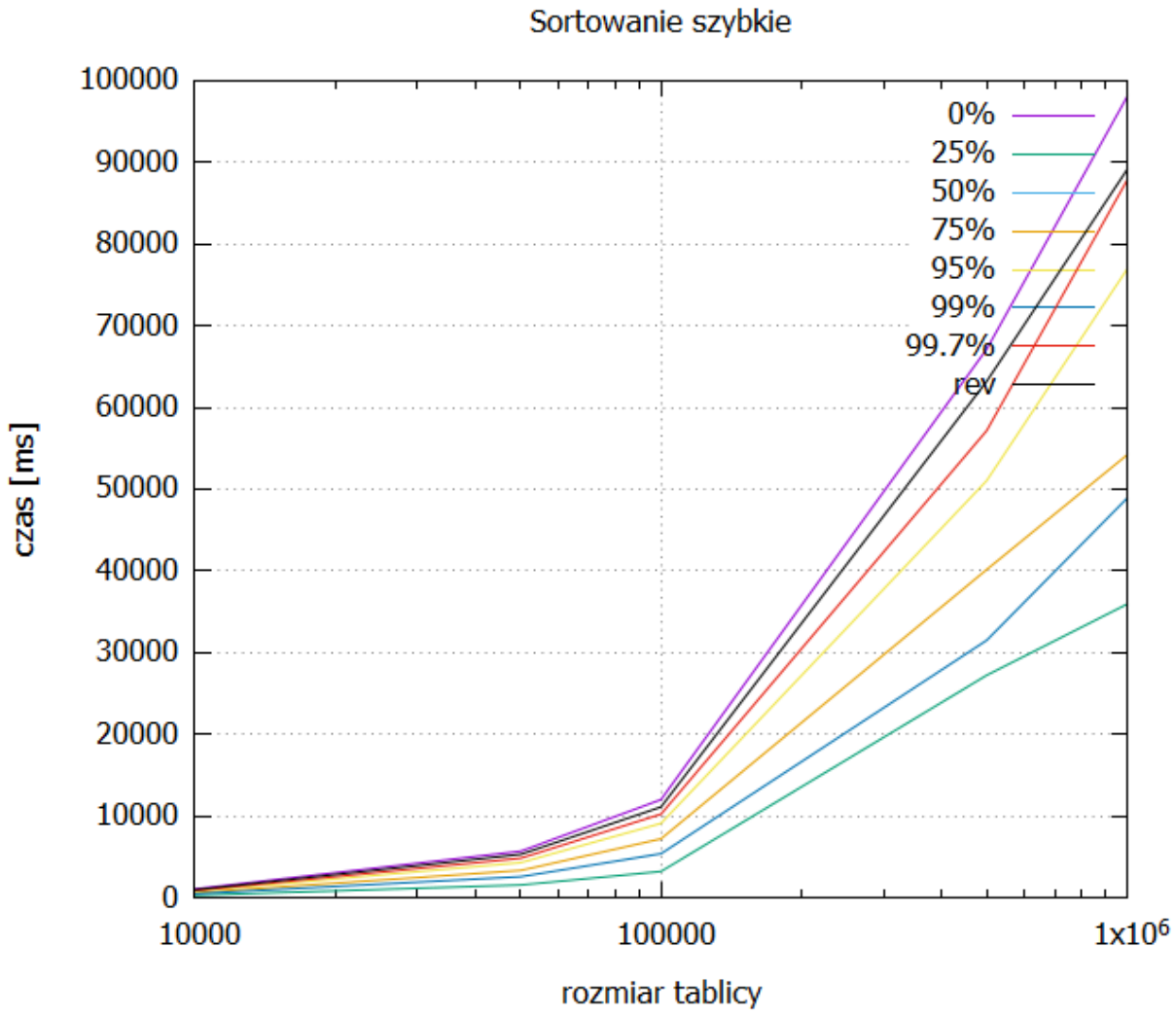
	10000	50000	100000	500000	1000000
0%	1730	9612	20056	111010	155403
25%	1165	6022	10250	10250	82752
50%	1157	6098	12640	15409	96897
75%	1159	5992	14519	65372	108804
95%	1184	6228	15409	69937	104129
99%	1288	6840	18117	78810	145312
99.7%	1496	8033	16212	89169	122187
odwrotnie	1621	8644	16412	99944	118323

Tabela 1:Wyniki pomiaru czasu sortowania przez scalanie dla stu tablic wyrażone w milisekundach



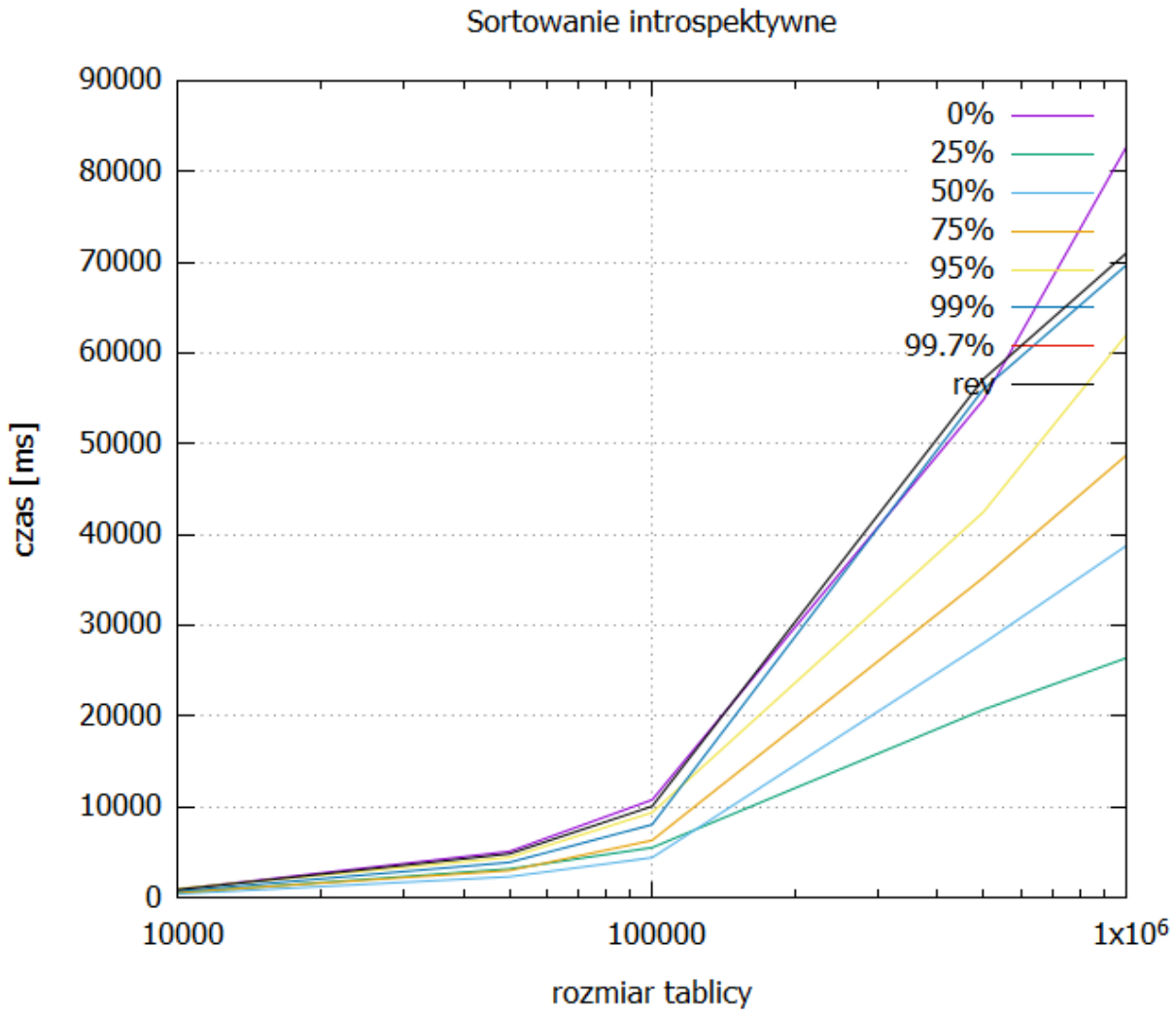
	10000	50000	100000	500000	1000000
0%	868	5677	11986	67148	98038
25%	277	1542	3195	27219	35926
50%	473	2558	5319	31292	46999
75%	476	2567	6330	31449	48880
95%	956	3317	7197	40163	54185
99%	1052	4278	10206	57167	89108
99.7%	599	4815	9084	11092	87817
odwrotnie	792	5293	11092	51051	76930

Tabela 2: Wyniki pomiaru czasu szybkiego sortowania dla stu tablic wyrażone w milisekundach



	10000	50000	100000	500000	1000000
0%	979	5073	10756	54852	82663
25%	398	2293	4393	20968	26389
50%	507	2612	5503	28021	48106
75%	536	2968	5504	31587	48713
95%	709	3154	6303	35248	61968
99%	905	4815	9358	42473	82663
99.7%	912	4415	8036	56014	70980
odwrotnie	876	3865	10055	57130	69663

Tabela 3: Wyniki pomiaru czasu sortowania introspektywnego dla stu tablic wyrażone w milisekundach



4 Podsumowanie i wnioski

- Najwolniejszy jest algorytm stosujący sortowanie przez scalanie.
- Sortowanie szybkie oraz introspektywne mają podobny czas pracy.
- W każdym przypadku najszybciej wykonywały się obliczenia dla tablicy posortowanej w 25%.
- Zazwyczaj obliczenia trwały najdłużej dla tablicy nieposortowanej, lecz w przypadku sortowania introspektywnego dla tablic mniejszych od 1000000 istnieje niewielka różnica w stosunku do tablic posortowanych w 90% i 99.7%.
- Obliczenia dla tablicy posortowanej odwrotnie w każdym przypadku trwały długo.
- Złożoność obliczeniowa dla sortowania przez scalanie nie wykazuje większych zmian wraz ze wzrostem rozmiaru tablic, co potwierdza oczekiwaną złożoność obliczeniową $O(nlog_2n)$.
- Złożoność obliczeniowa dla sortowania szybkiego również nie wykazuje większych zmian wraz ze wzrostem rozmiaru tablic, co potwierdza oczekiwaną złożoność obliczeniową $O(nlog_2n)$. Nie zauważono złożoności obliczeniowej $O(n^2)$.
- Złożoność obliczeniowa dla sortowania introspektywnego zmienia się jedynie dla tablic posortowanych w znacznym stopniu oraz przy dużych rozmiarach. Złożoność obliczeniowa w normalnym przypadku to $O(nlog_2n)$.

5 Literatura

1. Grębosz J., Symfonia C++ standard
2. Goodrich M., Data Structures and Algorithms in C++
3. Cormen T., Wprowadzenie do algorytmów
4. Wikipedia