

Data: 26.03.2020

Imię i nazwisko: Janusz Domaradzki

Nr albumu: 249024

Dane prowadzącego: mgr inż. Marta Emirsajłow

Termin zajęć: Piątek 13:15-14:45

Projektowanie Algorytmów i Metody Sztucznej Inteligencji

Projekt 1 – Algorytmy sortowania

1. Wprowadzenie

Celem projektu była analiza trzech algorytmów sortowania pod względem ich wydajności, tj. w jakim czasie są one w stanie posortować określone dane.

Testowi podlegały trzy algorytmy: **merge sort**, **quick sort** i **introspective sort**.

Test był przeprowadzany dla 100 tablic o rozmiarach:

- 10000
- 50000
- 100000
- 500000
- 1000000

Ponadto, testu sortowań dokonano dla poszczególnych przypadków:

- wszystkie elementy tablicy losowe
- 25%, 50%, 75%, 95%, 99%, 99,7% początkowych elementów tablicy jest już posortowanych
- wszystkie elementy tablicy już są posortowane, ale w odwrotnej kolejności

Każdy z algorytmów był sprawdzany dla tej samej tablicy tego samego przypadku (tj. na jednej tablicy dokonano sortowania 3 algorytmów dla jednego rozmiaru, dla jednego przypadku).

2. Opis algorytmów

a) Merge sort

Sortowana tablica dzielona jest rekurencyjnie na dwie podtablice aż do uzyskania tablic jednoelementowych. Następnie podtablice te są scalane w odpowiedni sposób, dający w rezultacie tablicę posortowaną. Wykorzystana jest tu metoda podziału problemu na mniejsze, łatwiejsze do rozwiązania zadania („dziel i rządź”).

Złożoność obliczeniowa:

- Najgorszy przypadek $O(n \log n)$
- Średni przypadek $O(n \log n)$

b) Quick sort

Na początku wybierany jest tzw. element osiowy. Następnie tablica dzielona jest na dwie podtablice. Pierwsza z nich zawiera elementy mniejsze od elementu osiowego, druga elementy większe lub równe, element osiowy znajdzie się między nimi. Proces dzielenia powtarzany jest aż do uzyskania tablic jednoelementowych, nie wymagających sortowania. Właściwe sortowanie jest tu jakby ukryte w procesie przygotowania do sortowania. Wybór elementu osiowego wpływa na równomierność podziału na podtablice (najprostszy wariant – wybór pierwszego elementu tablicy – nie sprawdza się w przypadku, gdy tablica jest już prawie uporządkowana).

Złożoność obliczeniowa:

- Najgorszy przypadek $O(n^2)$
- Średni przypadek $O(n \log n)$

c) Introspective sort

Jest to metoda hybrydowa, będąca połączeniem sortowania szybkiego i sortowania przez kopcowanie. Sortowanie introspektywne pozwala uniknąć najgorszego przypadku dla sortowania szybkiego (nierównomierny podział tablicy w przypadku, gdy jako element osiowy zostanie wybrany element najmniejszy lub największy).

Złożoność obliczeniowa:

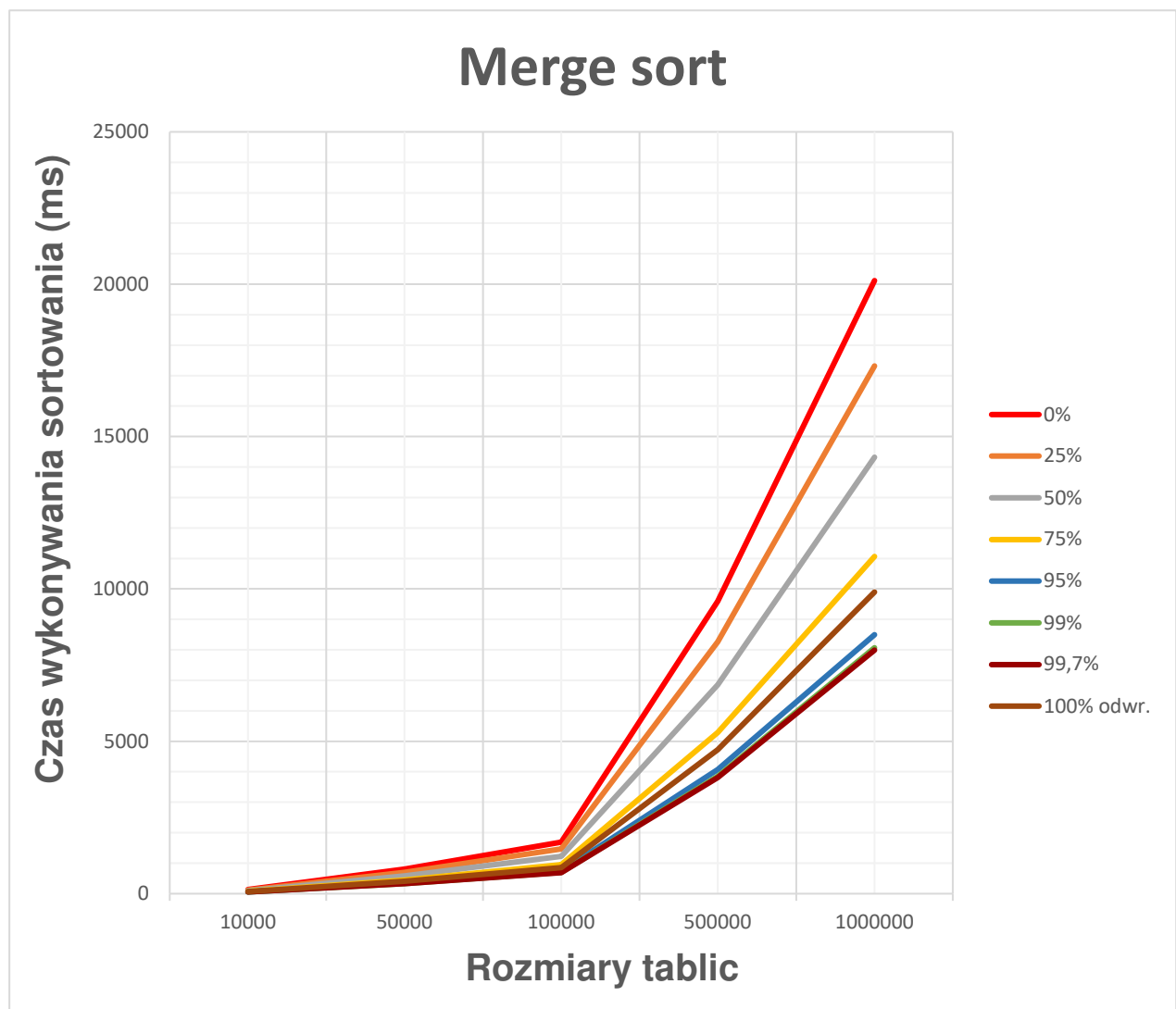
- Najgorszy przypadek $O(n \log n)$
- Średni przypadek $O(n \log n)$

3. Wyniki testu

Czasy poszczególnych sortowań zostały zapisane w milisekundach.

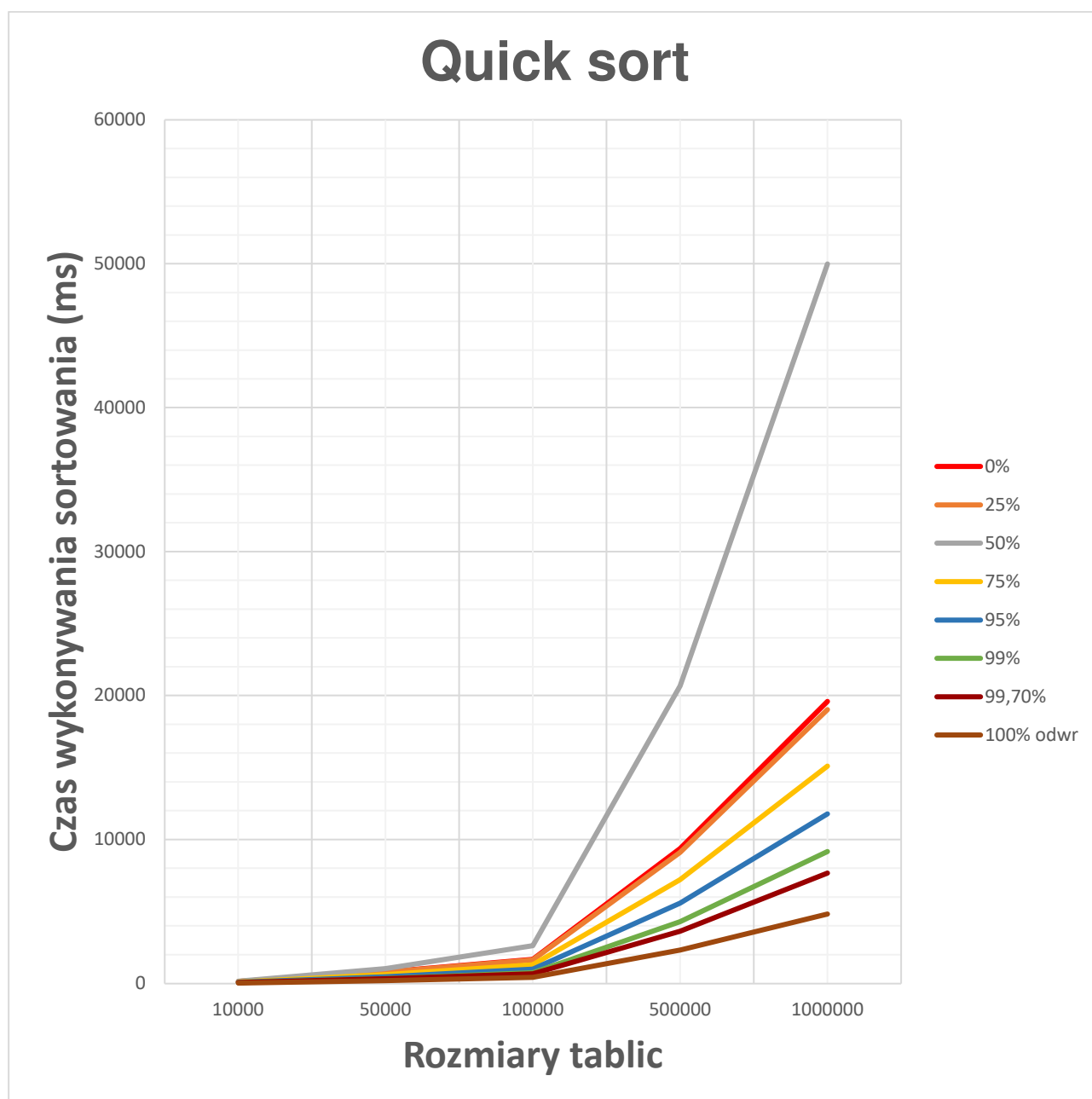
a) Merge sort

Merge sort	10000	50000	100000	500000	1000000
0%	141,305	801,542	1693,04	9600,53	20119
25%	120,186	694,667	1466,65	8266,73	17320,9
50%	100,753	577,507	1218,78	6857,62	14324,3
75%	78,7182	448,507	947,348	5287,42	11069
95%	61,7305	351,47	739,096	4068,96	8503,5
99%	58,6969	333,691	700,557	3848,45	8073,93
99,70%	58,3315	329,543	692,825	3809,74	7998,12
100 % odwr.	70,4608	405,092	853,434	4724,93	9901,03



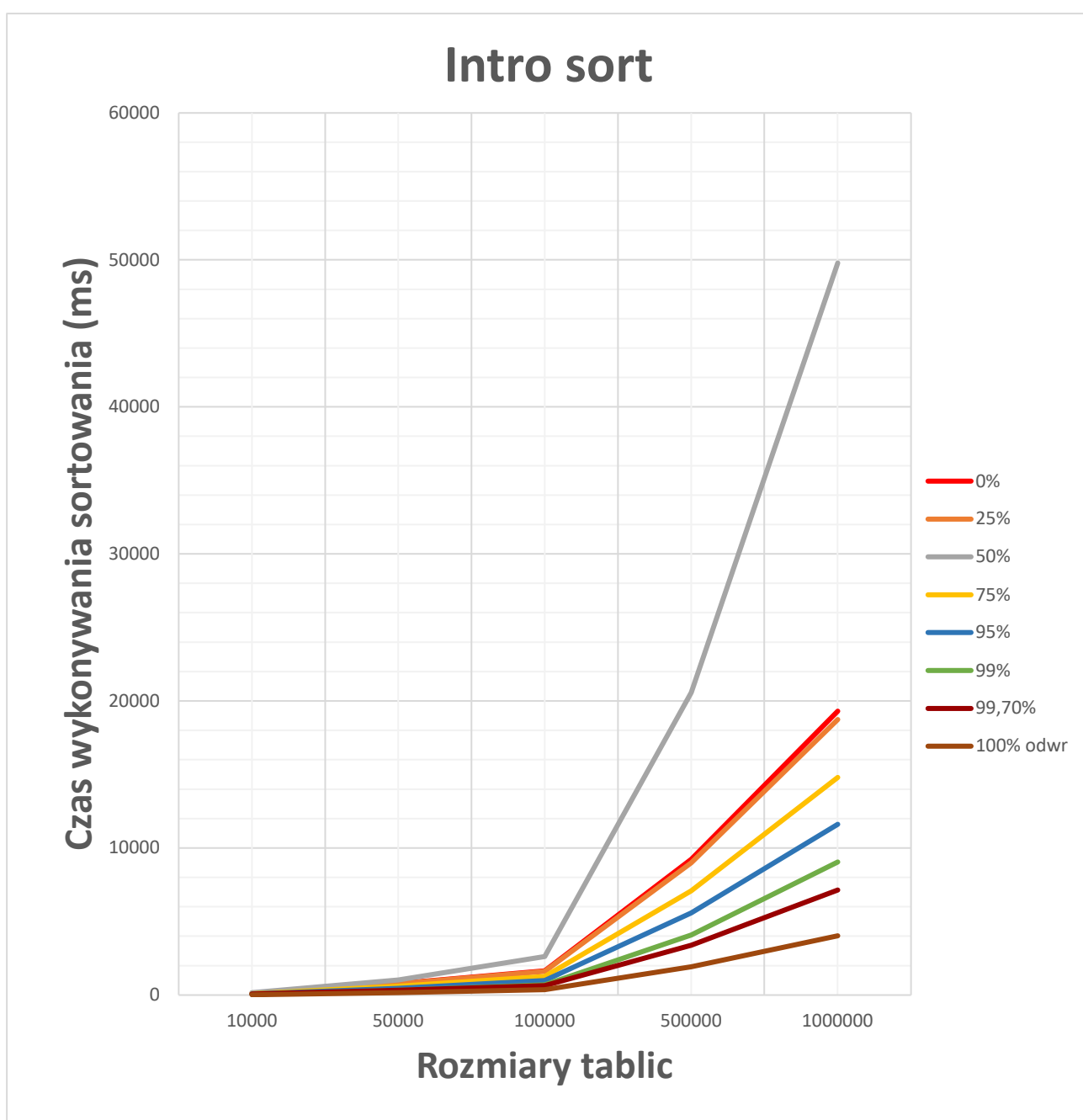
b) Quick sort

Quick sort	10000	50000	100000	500000	1000000
0%	139,769	794,778	1674,75	9370,1	19595,6
25%	135,884	774,496	1629,9	9108,3	19037
50%	165,125	1027,63	2616,53	20667,4	49997,6
75%	106,84	609,5	1285,46	7216,03	15107,6
95%	79,1414	466,584	994,034	5585,11	11790,6
99%	63,3833	348,804	738,893	4284,39	9177,6
99,70%	59,6647	320,328	659,245	3631	7668,96
100% odwr.	36,0318	197,976	415,575	2313,18	4828,26



c) Introspective sort

Intro sort	10000	50000	100000	500000	1000000
0%	137,217	782,266	1649,86	9231,67	19315,2
25%	133,627	761,849	1603,65	8983,96	18748
50%	163,122	1016,69	2614,04	20565	49790,5
75%	104,324	595,21	1256,99	7077,35	14801,6
95%	72,8231	448,748	982,988	5583,13	11623,7
99%	65,2554	328,635	688,761	4086,99	9045,32
99,70%	70,8581	325,463	643,976	3379,21	7142,25
100% odwr.	30,669	168,208	355,136	1917,44	4037,2



4. Wnioski

Dla przypadku, gdy 50% początkowych elementów tablicy jest już posortowane, sortowanie algorytmem szybkim i introspektywnym trwa o wiele dłużej, niż dla pozostałych przypadków. Dzieje się tak, ponieważ element osiowy wybrany jest jako element środkowy. W przypadku, gdy pierwsza połowa tablicy jest już posortowana, wartość ta jest wysoka.

Im większy procent posortowanych początkowo elementów, tym bardziej skraca się czas trwania algorytmu, ze względu na mniejszą liczbę potrzebnych obliczeń i porównań.

Złożoność obliczeniowa algorytmu sortowania szybkiego dla przypadku 50% początkowych elementów tablicy już posortowanych odpowiada charakterystyce kwadratowej, co potwierdza założenie najgorszego przypadku złożoności obliczeniowej, tj. $O(n^2)$.

5. Literatura

https://pl.wikipedia.org/wiki/Sortowanie_przez_scalanie

https://pl.wikipedia.org/wiki/Sortowanie_szybkie

https://pl.wikipedia.org/wiki/Sortowanie_introspektywne

https://pl.wikipedia.org/wiki/Sortowanie_przez_kopcowanie

Cormen T., Leiserson C.E., Rivest R.L., Stein C., Wprowadzenie do algorytmów, WNT