

Autor: Hubert Kowalczyk 259550	Struktury Danych i złożoność obliczeniowa Semestr letni 2022/2023	Termin: Wtorek NP: 17:05
Prowadzący: Dr. Inż. Tomasz Kapłon	Ćwiczenie 2	Data wykonania ćwiczenia: 28.03.2023
		Data oddania sprawozdania 25.04.2023

1. Cel ćwiczenia

Celem ćwiczenia było napisanie programu realizującego różne warianty sortowania. Podczas wykonywania ćwiczenia pierwszego napisano algorytm sortowania koktajlowego o złożoności $O(n^2)$. Podczas wykonywania ćwiczenia drugiego należało napisać algorytm o złożoności $O(n)$ oraz $O(n \log n)$. Z zaproponowanych algorytmów użyto algorytm Sortowania Przez kopcowanie $O(n \log n)$ oraz sortowanie przez zliczanie $O(n)$.

2. Wykonanie ćwiczenia

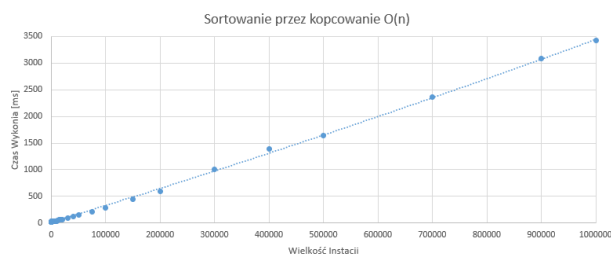
Program realizujący wymagane algorytmy sortowania został napisany przy pomocy edytora Clion na komputerze z systemem operacyjnym Windows 10. Program zrealizowano też z pomocą szkieletu napisanego na potrzeby ćwiczenia 1. Procesor komputera wykorzystanego podczas badania to Intel Core i7-9750H. Poza sortowaniem koktajlowym, przez kopcowanie oraz przez zliczanie program pozwala na sortowanie za pomocą: sortowanie bąbelkowego, sortowania przez wstawianie, sortowania przez wybieranie, algorytmu bogo sort, sortowanie szybkie, sortowanie grzebieniowe.

3. Sposób wykonania pomiarów

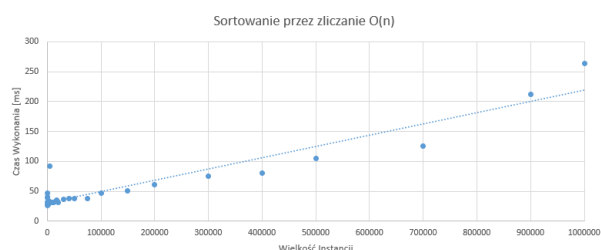
Dla każdego algorytmu badano złożoność obliczeniową. Dla różnych wielkości instancji badano czas trwania wykonania algorytmu. Dla każdej badanej wielkości instancji pomiar wykonany był dziesięciokrotnie. Następnie obliczano średnią z tych pomiarów. Średnia ta jest traktowana jako czas wykonywania algorytmu dla danej wielkości instancji. Wielkości instancji to: 10, 20, 30, 40, 50, 100, 150, 200, 350, 500, 1000, 1500, 2000, 3000, 4000, 5000, 7500, 10000, 12000, 15000, 17000, 20000, 30000, 40000, 50000, 75000, 100000, 150000, 200000, 300000, 400000, 500000, 700000, 900000, 1000000. Średnie wyniki czasowe wykonywania oraz wielkości instancji zgromadzono w tabeli 1 oraz utworzono na podstawie pomiarów wykresy. Wykresy te przedstawiono pod tabelą 1 na wykresach 1, 2.

Tabela 1 Wyniki pomiarów algorytmów Sortowania

Czas wykonania sortowania[ms]			Czas wykonania sortowania[ms]		
Wielkość instancji	Sortowanie przez kopcowanie	Sortowanie przez zliczanie	Wielkość instancji	Sortowanie przez kopcowanie	Sortowanie przez zliczanie
10	24,00	30,81	10000	32,99	30,81
20	23,22	31,79	12000	49,53	31,79
30	21,17	32,59	15000	56,39	32,59
40	23,29	35,77	17000	64,10	35,77
50	25,04	31,80	20000	70,46	31,80
100	23,08	36,06	30000	95,48	36,06
150	24,56	38,43	40000	119,82	38,43
200	22,69	37,79	50000	147,54	37,79
350	26,81	37,37	75000	217,22	37,37
500	24,97	50,25	100000	288,73	47,33
1000	31,24	61,22	150000	446,79	50,25
1500	26,87	75,74	200000	588,97	61,22
2000	27,58	80,99	300000	1007,27	75,74
3000	30,44	105,53	400000	1391,55	80,99
4000	31,93	124,98	500000	1638,93	105,53
5000	36,33	211,79	700000	2356,48	124,98
7500	31,62	264,37	900000	3087,30	211,79
10000	32,99	30,81	1000000	3423,44	264,37



Wykres 1: Złożoność czasowa algorytmu sortowania przez kopcowanie.



Wykres 1: Złożoność czasowa algorytmu sortowania przez zliczanie.

4. Wnioski

Na podstawie wyników przeprowadzonych pomiarów złożoności czasowej stwierdzono że:

-Wykres złożoności czasowej sortowania przez kopcowanie przypomina wykres funkcji $O(n) = n \log n$ lub $O(n)$.

-Wykres złożoności czasowej sortowania przez zliczanie przypomina wykres funkcji $O(n) = n$.

-Mimo teoretycznie mniejszej złożoności obliczeniowej sortowanie przez kopcowanie sortowało kilkakrotnie dłużej instancje o dużej wielkości.

-Oba sortowanie mają bardzo podobne i nie zmieniające się czasy wykonywania dla wielkości instancji $n < 10000$. Prawdopodobnie spowodowane jest to bardzo szybkim wykonaniem algorytmu i ciągłą pracą systemu operacyjnego.

-Sortowanie przez kopcowanie dla swojego działania potrzebuje struktury jaką jest kopiec.

-Sortowanie przez kopcowanie wykonuje się dłużej zwłaszcza dla wielkości instancji $n > 10000$. Być może jest to spowodowane potrzebą utworzenia oraz pobierania elementu z struktury danych jaką jest kopiec.

