

Data laboratorium: 04.03.2016

PAMSI

Sprawozdanie z laboratorium nr 1

Tablice dynamiczne, złożoność
obliczeniowa

PAWEŁ JAGODZIŃSKI 218496

1. Cele laboratorium:

- a) Zaznajomienie się z pojęciem złożoności obliczeniowej algorytmów,
- b) Tworzenie tablic dynamicznych i ich wykorzystanie.

2. Opis ćwiczenia

Należało wykonać eksperyment, polegający na zmierzeniu czasu potrzebnego do wypełnienia tablicy o rozmiarze początkowym N_0 , odpowiednio: $10, 10^2, 10^3, 10^6, 10^9$ elementami.

$N_0 = 10$ elementów.

3. Wyniki eksperymentu

a) Wyniki dla komputera z procesorem dwurdzeniowym 2x2,0 GHz (intel CORE i3)

		metoda 1	metoda 2	metoda 3
ilość el.	pomiar nr	t [μs]	t [μs]	t [μs]
10	1.	62	113	48
10	2.	52	111	116
10	3.	112	48	55
10	4.	124	55	111
10	5.	111	136	112
10	6.	52	60	50
10	7.	59	112	49
10	8.	129	49	118
10	9.	61	64	48
10	10.	110	114	62
czas średni		87,2	86,2	76,9
100	1.	119	63	53
100	2.	260	52	60
100	3.	300	115	51
100	4.	118	112	111
100	5.	118	47	48
100	6.	118	48	111
100	7.	120	126	115
100	8.	122	112	53
100	9.	109	80	55
100	10.	108	55	112
czas średni		149,2	81	76,9

		metoda 1	metoda 2	metoda 3
ilość el.	pomiar nr	t [μs]	t [μs]	t [μs]
1000 = 10 ³	1.	10325	96	238
1000 = 10 ³	2.	4305	228	132
1000 = 10 ³	3.	9912	108	75
1000 = 10 ³	4.	4138	267	133
1000 = 10 ³	5.	4085	255	62
1000 = 10 ³	6.	9647	103	63
1000 = 10 ³	7.	5621	237	76
1000 = 10 ³	8.	4185	121	58
1000 = 10 ³	9.	4086	317	132
1000 = 10 ³	10.	10210	231	63
czas	średni	6651,4	196,3	103,2
100000 = 10 ⁵	1.	40901400	456237	732
100000 = 10 ⁵	2.	40803500	461045	1819
100000 = 10 ⁵	3.	41021800	447549	1816
100000 = 10 ⁵	4.	41089900	464847	1776
100000 = 10 ⁵	5.	40770100	457376	1818
100000 = 10 ⁵	6.	40952000	459749	793
100000 = 10 ⁵	7.	40911800	448558	1803
100000 = 10 ⁵	8.	40938300	447326	732
100000 = 10 ⁵	9.	40793300	449332	809
100000 = 10 ⁵	10.	41151200	447467	1767
czas	średni	40933330	453948,6	1386,5

		metoda 1	metoda 2	metoda 3
ilość el.	pomiar nr	t [μ s]	t [μ s]	t [μ s]
$1000000 = 10^6$	1.	4254080000	47466700	12975
$1000000 = 10^6$	2.	4249070000	48063800	6309
$1000000 = 10^6$	3.	4248140000	48063800	6352
$1000000 = 10^6$	4.	4249960000	48063800	13240
$1000000 = 10^6$	5.	4248640000	48063800	8331
$1000000 = 10^6$	6.	4243130000	48063800	15920
$1000000 = 10^6$	7.	4243310000	47637000	6620
$1000000 = 10^6$	8.	4246710000	47807800	14847
$1000000 = 10^6$	9.	4245080000	47721500	15590
$1000000 = 10^6$	10.	4248870000	47949600	10256
czas	średni	4247699000	47890160	11044
$1000000000 = 10^9$	1.	-	-	6925990
$1000000000 = 10^9$	2.	-	-	6606490
$1000000000 = 10^9$	3.	-	-	6405340
$1000000000 = 10^9$	4.	-	-	6435720
$1000000000 = 10^9$	5.	-	-	6435720
$1000000000 = 10^9$	6.	-	-	6311600
$1000000000 = 10^9$	7.	-	-	6482880
$1000000000 = 10^9$	8.	-	-	6373090
$1000000000 = 10^9$	9.	-	-	6373090
$1000000000 = 10^9$	10.	-	-	6844000
czas	średni	-	-	6519392

Omówienie tabeli z wynikami eksperymentu:

- **ilość el.** - oznacza ilość elementów tablicy,
- **t [μ s]** – czas potrzebny do wypełnienia tablicy zadany rozmiarem, mierzony w mikrosekundach 10^{-6} [s],
- **metoda 1, metoda 2, metoda 3-** trzy algorytmy tworzenia rozmiaru tablic.

Dla metody 2 nie mam wyników do tablic o rozmiarze 10^9 , ponieważ czas wypełniania tablicy tym algorytmem trwał ponad 14 h ciąglej pracy, a laptop potrzebny był mi na uczelnię oraz do celów naukowych, podobnie ma się sprawa z metodą 1. Czas trwania metody 1 byłby jeszcze dłuższy.

b)

Wyniki dla komputera z procesorem czterordzeniowym 4 x 2,2 GHz
(intel CORE i7)

ilość el.	Nr Pomiaru	metoda 1	metoda 2	metoda 3
		t [μs]	t [μs]	t [μs]
10	1.	118	158	107
10	2.	135	166	69
10	3.	133	137	143
10	4.	107	94	186
100 = 10 ²	1.	109	140	165
100 = 10 ²	2.	319	127	213
100 = 10 ²	3.	106	103	141
100 = 10 ²	4.	242	138	283
1000 = 10 ³	1.	10603	232	138
1000 = 10 ³	2.	9298	274	199
1000 = 10 ³	3.	6422	212	166
1000 = 10 ³	4.	10557	240	171
100000 = 10 ⁵	1.	25924487	370974	1844
100000 = 10 ⁵	2.	26050401	310742	1674
100000 = 10 ⁵	3.	26091521	361020	1647
100000 = 10 ⁵	4.	25950000	394833	1802
1000000 = 10 ⁶	1.	2658010000	29383461	13066
1000000 = 10 ⁶	2.	2646890000	29601138	14707
1000000 = 10 ⁶	3.	2650670000	29537744	15269
1000000000 = 10 ⁹	1.	-	-	3745870
1000000000 = 10 ⁹	2.	-	-	3722241
1000000000 = 10 ⁹	3.	-	-	3783493

Tabela nr 1.

Omówienie tabeli z wynikami eksperymentu:

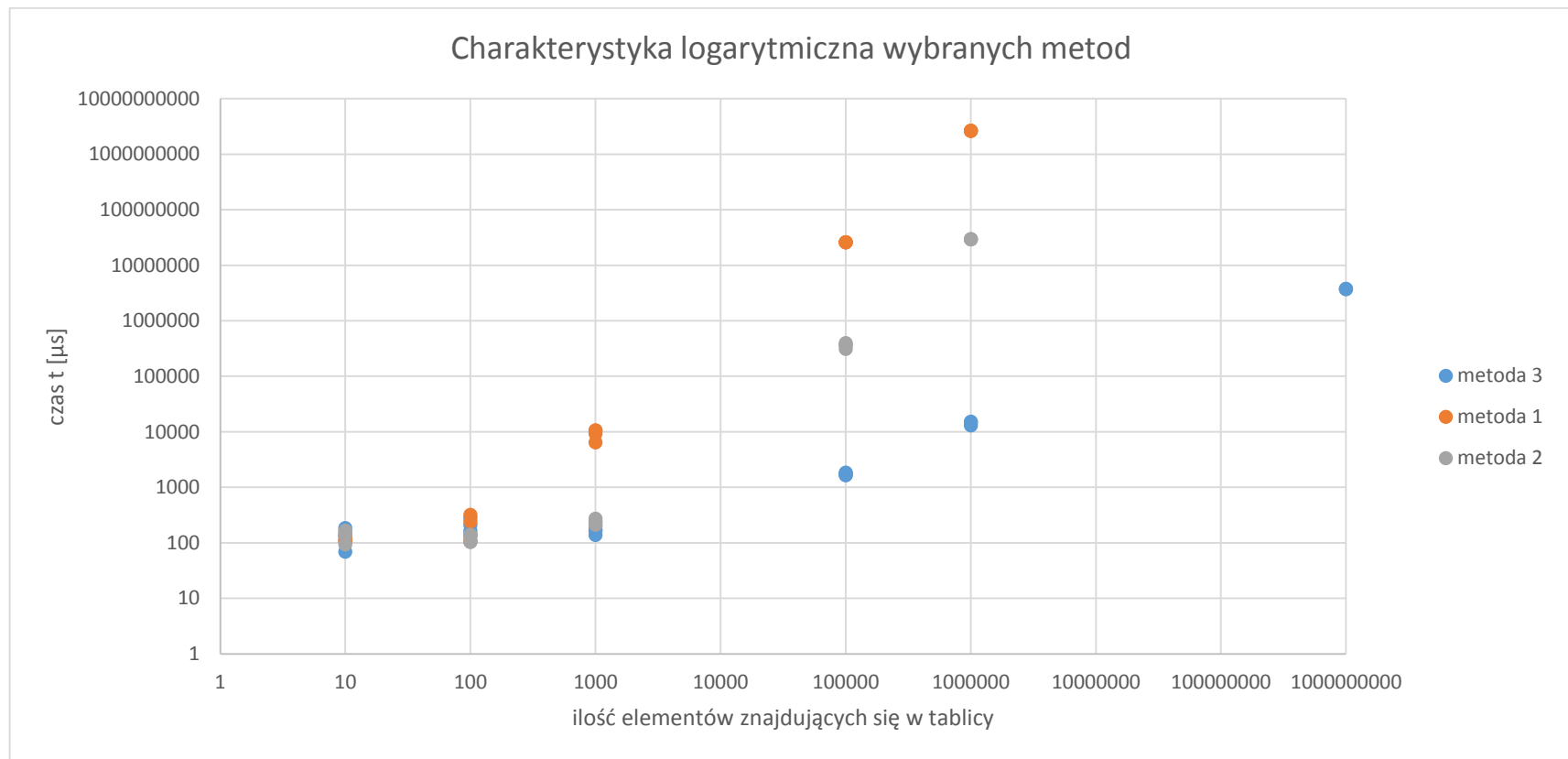
- **ilość el.** - oznacza ilość elementów tablicy,
- **t [μs]** – czas potrzebny do wypełnienia tablicy zadany rozmiarem, mierzony w mikrosekundach 10⁻⁶ [s],
- **metoda 1, metoda 2, metoda 3**- trzy algorytmy tworzenia rozmiaru tablic.

4. Opis algorytmów (idea):

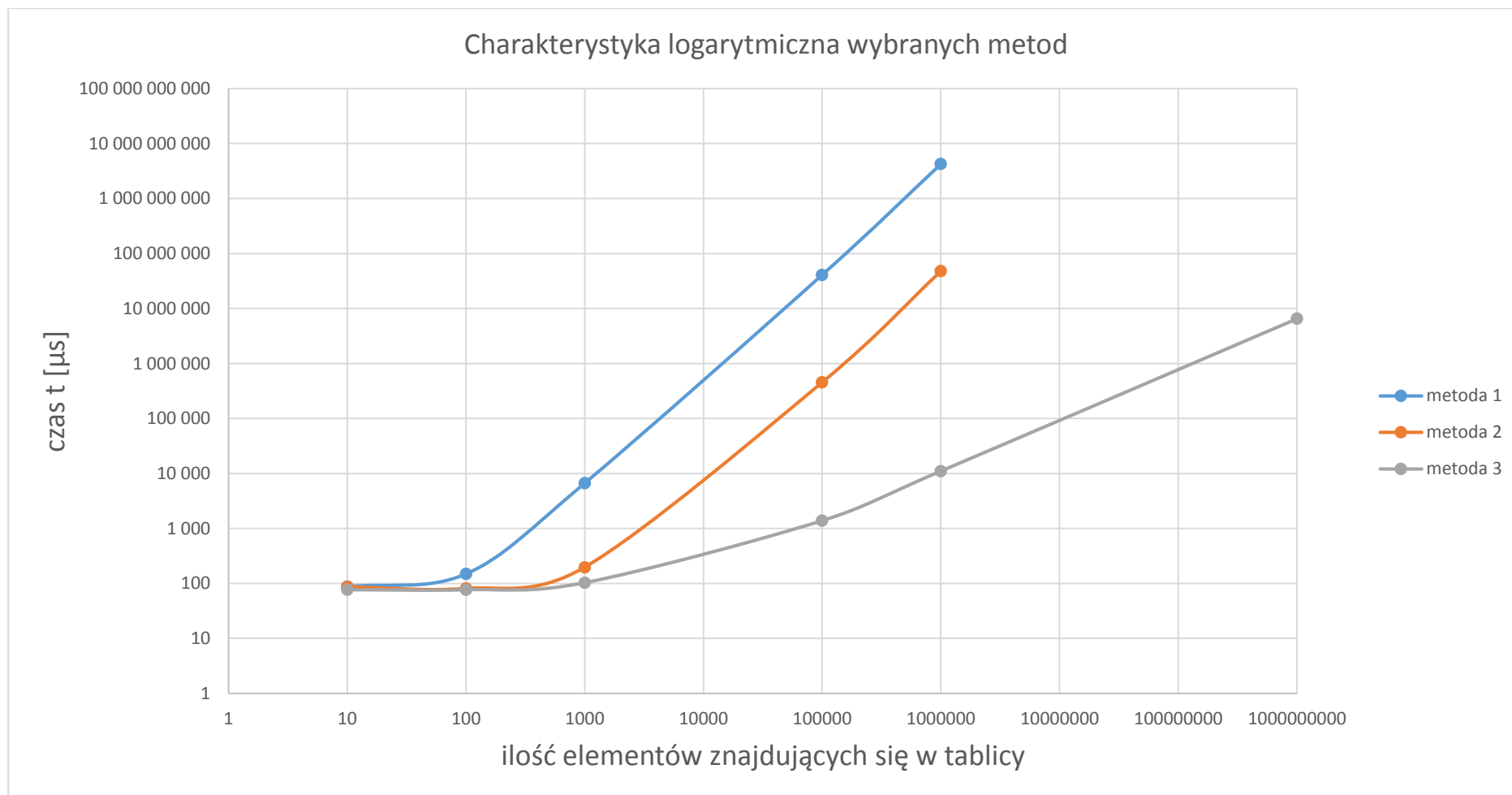
- **metoda 1** – tworzymy tablicę o rozmiarze początkowym $N_0=10$ elementów, gdy zostanie ona wypełniona w całości to tworzona jest nowa tablica o rozmiarze o 1 większym od poprzedniej. Cała zawartość tablicy początkowej kopiowana jest do nowo utworzonej. Dobierany jest jeden element w celu zapelnienia nowej tablicy. Operacje te są powtarzane, aż do powstania tablicy o oczekiwanym rozmiarze.
- **metoda 2** – tworzymy tablicę o rozmiarze początkowym $N_0=10$ elementów, gdy zostanie ona wypełniona w całości to tworzona jest nowa tablica o rozmiarze o 100 większym od poprzedniej. Cała zawartość tablicy początkowej kopiowana jest do nowo utworzonej. Dobierane są brakujące elementy do zapelnienia tablicy w całości. Operacje te są powtarzane, aż do powstania tablicy o oczekiwanym rozmiarze.
- **metoda 3** - tworzymy tablicę o rozmiarze początkowym $N_0=10$ elementów, gdy zostanie ona wypełniona w całości to tworzona jest nowa tablica o rozmiarze o 10 razy większym od poprzedniej. Cała zawartość tablicy początkowej kopiowana jest do nowo utworzonej. Dobierane są brakujące elementy do zapelnienia tablicy w całości. Operacje te są powtarzane, aż do powstania tablicy o oczekiwanym rozmiarze.

Implementacja podczas tworzenia 1 i 2 tablicy jest zmieniona w stosunku do pozostałych tzn. 3,4....n

Elementami tablicy są wartości zmiennych sterujących.



Rys. 1. Wykres przedstawiający zależność ilości elementów w tablicy do czasu potrzebnego na ich uzupełnienie trzema algorytmami. (intel CORE i7)



Rys. 2. Wykres przedstawiający zależność ilości elementów w tablicy do czasu potrzebnego na ich uzupełnienie trzema algorytmami. (intel CORE i3)

5. Wnioski

Porównując wykresy wybranych metod (Rys. 1, Rys. 2.) można dostrzec, że wypełniając tablice 10, 100 elementami czasy trwania wszystkich badanych algorytmów są niemal identyczne. Przy 1000 elementach wykonanie algorytmu metody 1, trwa ok. 100 razy dłużej (Rys. 2) od dwóch pozostałych. Dla 100 000 elementów każda charakterystyka ma inny przebieg. Algorytm 1 jest najwolniejszy, trzeci najszybszy. W przypadku 10^6 elementów, czasy trwania metod: 1,2,3 odbiegają znacznie od siebie.

Wykorzystując notację duże O metoda 3 przypomina swoim przebiegiem złożoność obliczeniową $O(n \log n)$.

Reasumując, w celu uzyskania odpowiedzi, który algorytm jest szybszy potrzebna jest nam duża liczba danych do przetworzenia np. w przypadku wypełnienia tablicy 10^9 elementami metoda 2 potrzebowała ponad 14h (czas nie jest dokładnie znany i myślę, że sporo dłuższy) przy czym algorytm 3 wykonał tę operację w odpowiednio w zależności od procesora 3,8 [s] – i7 oraz 6,5 [s] – i3.

Gdy nasze badania składają się z małej ilości danych nie warto pracować nad super szybkimi rozwiązaniami, bo sama implementacja pochłonie wiele czasu.

Do moich testów złożoności obliczeniowej wykorzystałem dwa komputery o różnych parametrach, wniosek nasuwa się taki, że w dzisiejszym świecie oprócz dobrego i wydajnego algorytmu potrzebny jest szybki komputer. Dla przykładu np. wypełniając tablicę 10^6 elementami metodą nr 2, komputer z procesorem i7 wykonał tę operację w ok 44 min przy czym komputer z procesorem i3 potrzebował, aż ok 71 min. Ten sam algorytm, ale znaczna różnica w czasie. Dla przypadku wypełniania tablicy 10^9 elementami różnica jest kolosalna.