

Sprawozdanie PAMSI

Zadanie: Przetestować czas dodawania nowych elementów na tablicę asocjacyjną w zależności od ilości elementów. Testy wykonać dla różnej ilości bucketów w tablicy.

Poprawione sprawozdanie.

W pierwszym sprawozdaniu dla pomiarów dodawania otrzymywałem wykresy różne od oczekiwanych. Było to spowodowane wypisywaniem w terminalu (`std::cout<<`), zawartości bucketa podczas dodawania, co znacząco wpłynęło na otrzymane pomiary.

Tab 1

Ilość bucketów: 26 + 1 bucket na niezidentyfikowane			
Ilość danych	Czas dodawania [ms]	Czas wyszukiwania [ms]	Ilość elementów w przeszukiwanym buckecie
10	0	0	2
	0	0	
	0	0	
	1	0	
	1	0	
	0	0	
	1	0	
	0	0	
	2	1	
	1	0	
100	1	1	6
	1	0	
	1	1	
	0	1	
	2	0	
	1	1	
	1	1	
	1	1	
	1	0	
	1	0	
1000	11	1	56
	4	0	
	7	1	
	6	1	
	4	1	
	4	1	
	7	0	
	8	1	
	6	1	
	5	1	
	4	1	
10000	36	1	631
	37	0	
	34	2	
	36	1	
	34	2	
	38	1	
	36	0	
	36	1	
	37	1	
	35	1	
100000	332	1	6652
	334	2	
	326	2	
	349	1	
	338	2	
	328	2	
	331	2	
	340	2	
	345	1	
	350	2	
1000000	1266	3	26003
	1257	3	
	1274	4	
	1263	3	
	1293	2	
	1242	3	
	1285	4	
	1274	5	
	1263	3	
	1288	3	

Tab 2

Ilość bucketów: 13 + 1 bucket na niezidentyfikowane			
Ilość danych	Czas dodawania [ms]	Czas wyszukiwania [ms]	Ilość elementów w przeszukiwanym buckecie
10	1	1	3
	1	0	
	0	1	
	0	0	
	1	1	
	0	1	
	1	0	
	0	1	
	0	1	
	1	0	
100	2	1	14
	0	0	
	1	1	
	0	1	
	1	2	
	0	1	
	1	1	
	2	1	
	2	0	
	2	0	
1000	4	0	118
	7	0	
	8	1	
	6	1	
	4	2	
	7	1	
	6	2	
	5	1	
	7	1	
	7	1	
	7	1	
10000	35	1	1314
	36	2	
	35	1	
	34	1	
	37	2	
	37	1	
	38	1	
	39	2	
	36	2	
	36	2	
100000	336	3	13053
	344	2	
	333	1	
	342	3	
	349	3	
	361	2	
	355	1	
	369	3	
	384	3	
	388	2	
1000000	1277	4	50653
	1238	4	
	1284	5	
	1285	4	
	1278	4	
	1289	3	
	1280	5	
	1279	4	
	1283	4	
	1292	4	

Tab 3

Ilość bucketów: 2 + 1 bucket na niezidentyfikowane			
Ilość danych	Czas dodawania [ms]	Czas wyszukiwania [ms]	Ilość elementów w przeszukiwanym buckecie
10	0	0	6
	0	0	
	1	1	
	2	0	
	0	1	
	0	0	
	1	0	
	0	0	
	1	1	
	0	2	
100	1	1	58
	1	0	
	2	1	
	1	1	
	0	2	
	1	1	
	1	1	
	1	1	
	2	0	
	1	0	
1000	4	1	562
	7	2	
	8	1	
	12	1	
	4	2	
	7	1	
	6	1	
	6	1	
	7	1	
	6	0	
	4	1	
10000	36	1	5782
	38	2	
	36	1	
	35	1	
	35	2	
	37	1	
	40	1	
	34	2	
	38	2	
	36	2	
100000	357	3	57955
	396	3	
	354	3	
	381	3	
	389	2	
	353	3	
	364	3	
	371	3	
	335	2	
	364	3	
1000000	1274	9	224668
	1291	9	
	1286	8	
	1278	9	
	1263	8	
	1259	7	
	1287	8	
	1274	8	
	1263	9	
	1289	9	

Uśrednione wyniki:

26 bucketów		
Wartość średnia		
Ilość danych	Czas dodawania [ms]	Czas wyszukiwania [ms]
10	0,60	0,10
100	1,00	0,60
1000	6,20	0,80
10000	35,90	1,00
100000	337,30	1,70
1000000	1270,50	3,30

Tab 1.1

13 bucketów		
Wartość średnia		
Ilość danych	Czas dodawania [ms]	Czas wyszukiwania [ms]
10	0,50	0,60
100	1,10	0,80
1000	6,10	1,00
10000	36,30	1,50
100000	356,10	2,30
1000000	1278,50	4,10

Tab 2.1

2 buckety		
Wartość średnia		
Ilość danych	Czas dodawania [ms]	Czas wyszukiwania [ms]
10	0,50	0,50
100	1,10	0,80
1000	6,70	1,10
10000	36,50	1,50
100000	366,40	2,80
1000000	1276,40	8,40

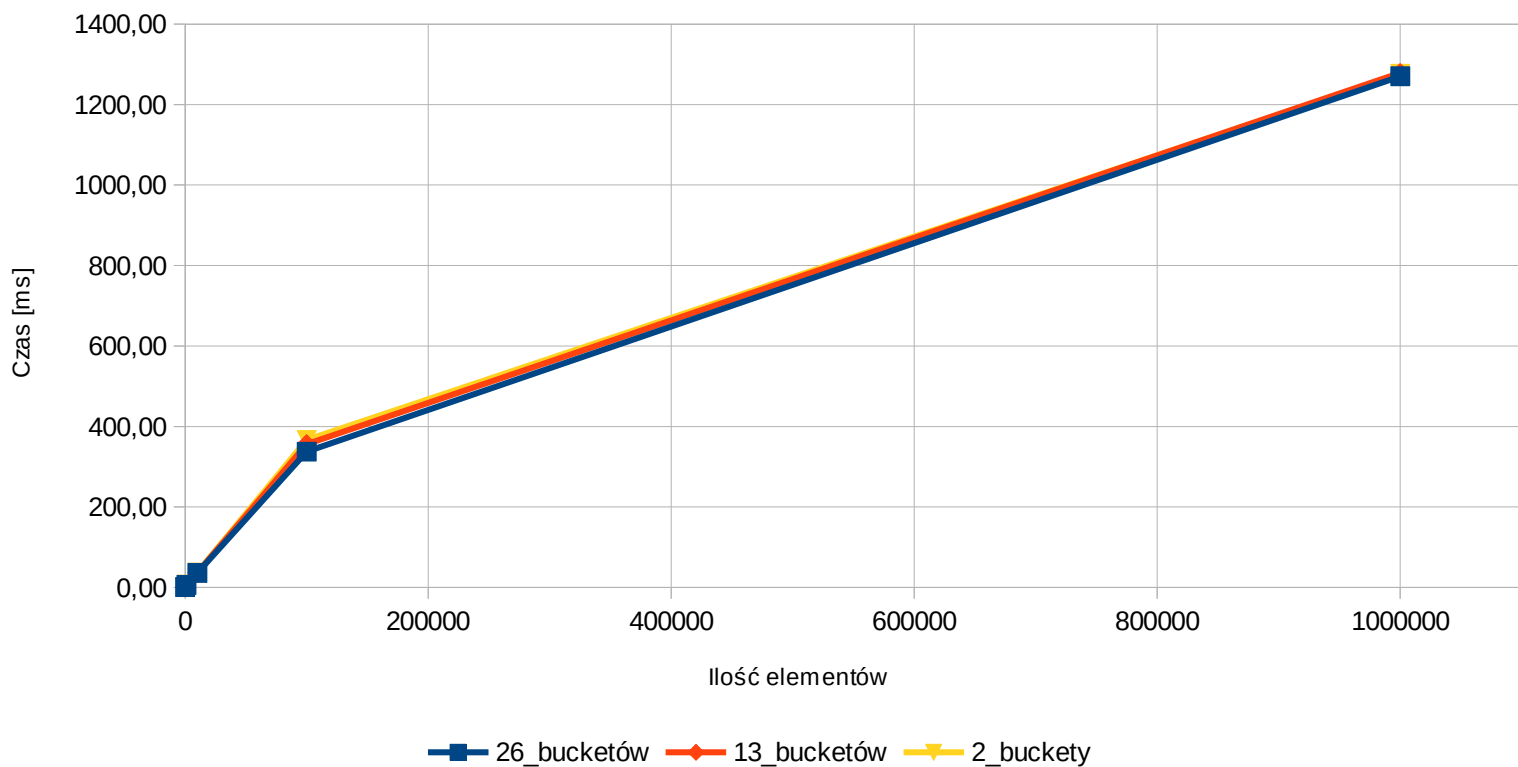
Tab 3.1

Zmiana pojemności bucketów		
Tabela 1	Tabela 2	Tabela3
2	3	6
6	14	58
56	118	562
631	1314	5782
6652	13053	57955
26003	50653	224668

Tab 4

Średnie czasy dodawania

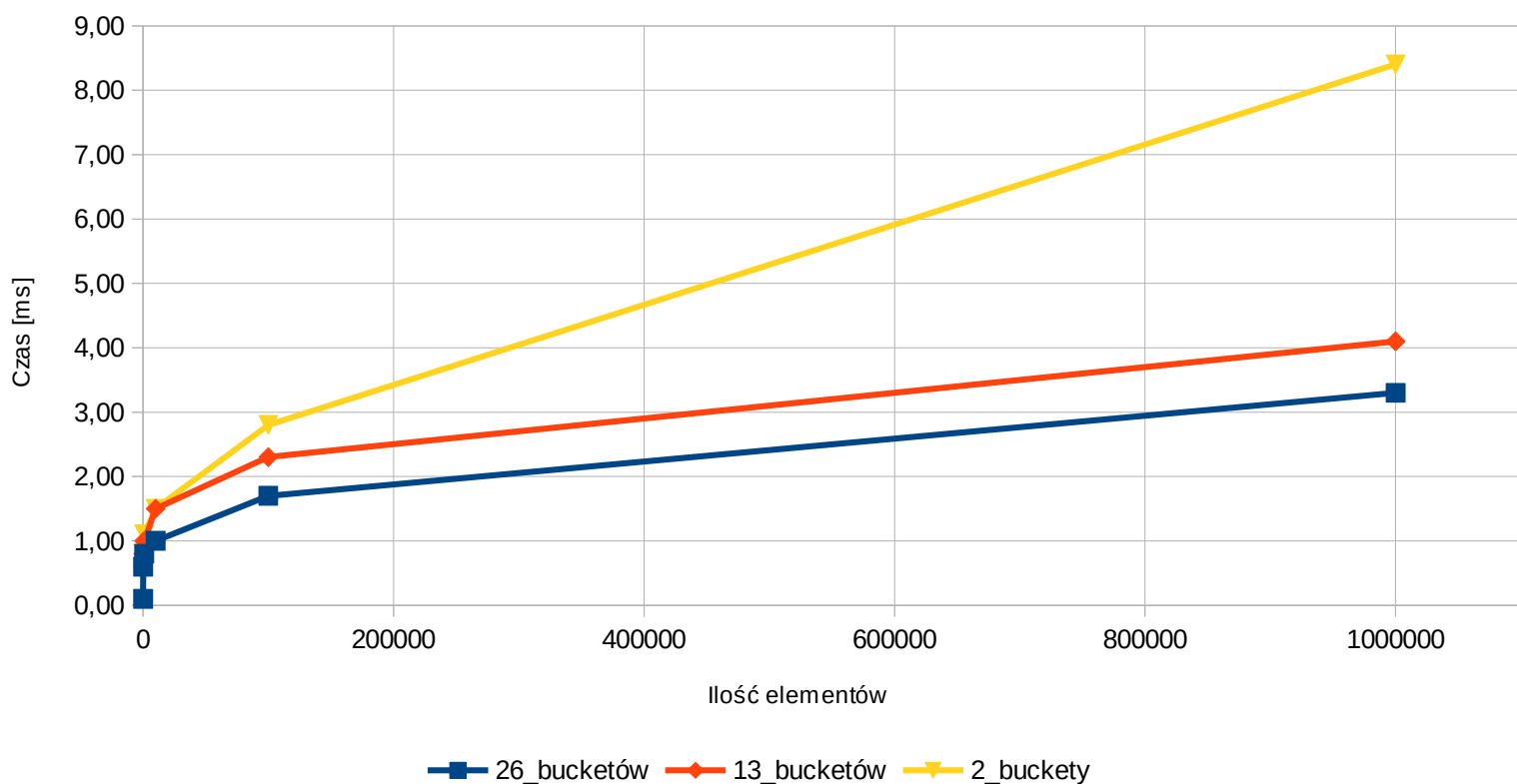
Zestawienie średnich czasów dodawania dla różnej ilości bucketów



Wykres 1

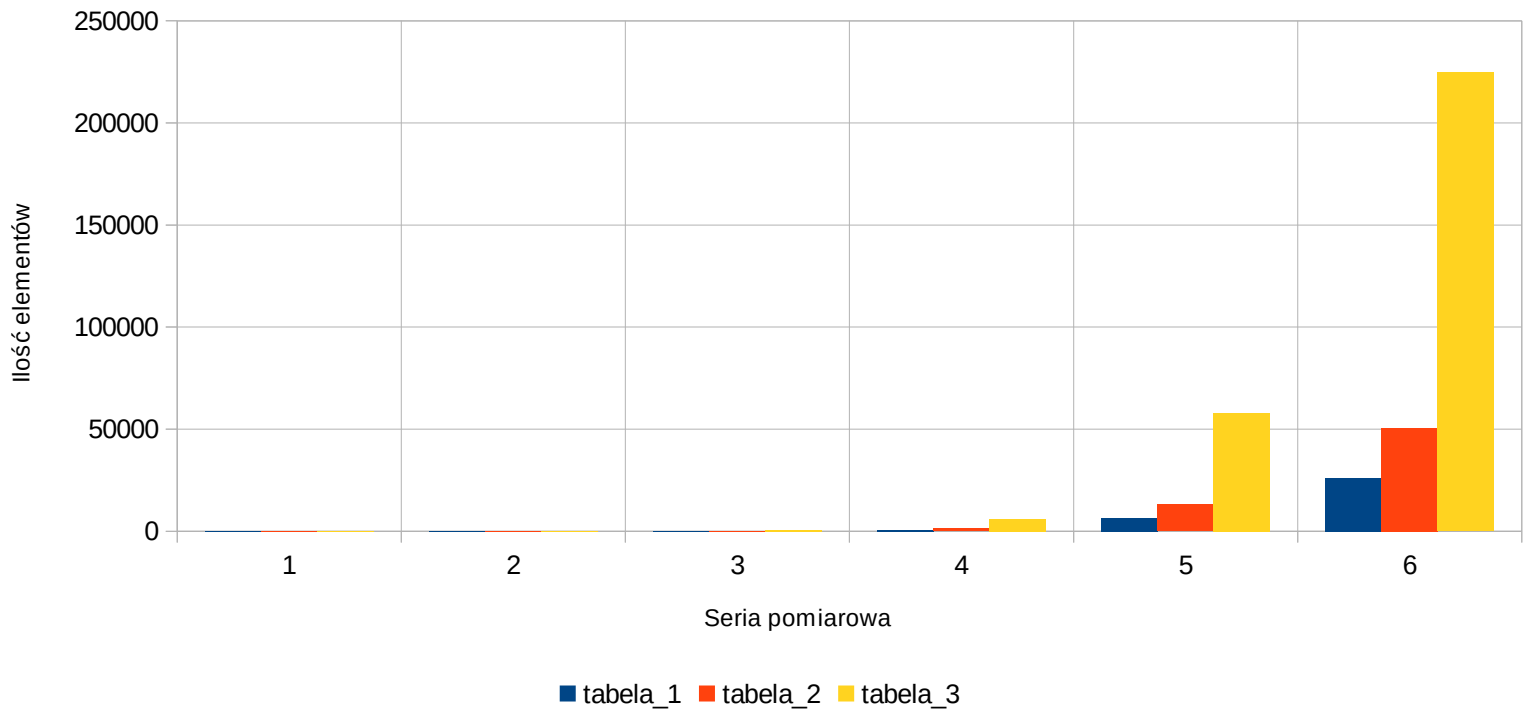
Średnie czasy wyszukiwania

Zestawienie średnich czasów wyszukiwania dla różnej ilości bucketów



Wykres 2

Zmiana pojemności przeszukiwanych bucketów



Wykres 3

Wnioski:

Zgodnie z wynikami odczytanymi z *Wykres 1*, opisującego zmiany czasu dodawania elementów w zależności od ilości elementów do dodania, charakterystyka złożoności obliczeniowej wydaje się być liniowa $O(n)$. Taka złożoność wynika z tego, że każdy element dodawany jest w następującej sekwencji: wczytaj, odszyfruj hasz elementu, umieść w odpowiednim polu tablicy **na początku listy**. Wstawianie na początek listy unika zbędnego „przechodzenia” po wskaźnikach listy i dla każdego elementu wymaga tyle samo czasu. Pokrywające się wykresy mówią nam o tym, że zgodnie z oczekiwaniami liczba bucketów nie ma wpływu na czas dodawania.

Wyniki odczytane z *Wykres 2*, opisującego zmiany czasu przeszukiwania listy w zależności od ilości dodanych elementów, są zbyt mało miarodajne aby móc na ich podstawie oszacować złożoność obliczeniową algorytmu. Z tego wykresu możemy odczytać jednak jedną prawidłowość. Wraz ze wzrostem ilości elementów w buckecie zwiększa się czas przeszukiwania go w celu znalezienia elementu umiejscowionego na końcu bucketa. Spowodowane jest to koniecznością przeszukiwania w następującej sekwencji: (od początku listy) sprawdź czy wskazany element jest poszukiwanym, jeżeli nie → sprawdź następny element. Taka sekwencja wiąże się z koniecznością przechodzenia po wskaźnikach, do momentu znalezienia wyszukiwanego elementu, lub dojścia do końca listy, co zwiększa czas przeszukiwania w zależności od ilości elementów do przejścia. A więc wykres odpowiada oczekiwanemu rezultatowi.

Wyniki z *Wykres 3* są logicznie poprawne. Wraz ze zmniejszeniem ilości bucketów, będą one przechowywały większą ilość danych.

Na bazie otrzymanych wyników możemy wywnioskować, że uzyskanie najlepszej złożoności obliczeniowej $O(1)$ może zostać osiągnięte przez zwiększenie ilości bucketów, czyli przyporządkowania każdemu dodanemu elementowi unikalnego hashu. Wymagało by to jednak unikalnego miejsca w tablicy asocjacyjnej dla każdego elementu, co znacznie zwiększyło by rozmiar tablicy.