Sprawozdanie - Laboratorium 06 PAMSI

Artur Gasiński — 218685

11.04.2016

1 Zadanie

1. Implementacja tablicy asocjacyjnej w postaci tablicy z haszowaniem.
2. Dobór i implementacja funkcji skrótu, obliczającej indeks tablicy na podstawie podanego klucza (łańcucha znaków).
3. Pomiar czasów:
   - wypełnienia tablicy z haszowaniem dla różnych liczb elementów do zapisania:
   $$n = 10, 10^2, 10^3, 10^6, 10^7; \quad (1)$$
   - odczytu pojedynczego elementu o wybranym kluczu z tablicy w zależności od liczby elementów n w tablicy.

2 Wykonanie

1. Struktura programu:
   - interfejs IAssotiative (zawiera podstawowe metody ADT tablicy asocjacyjnej),
   - klasa SList, zawierająca metody zarządzania listą (przechowuje węzły z danymi postaci: klucz-wartość),
   - klasa HashTab, zawierająca implementację tablicy z haszowaniem:
     - tablica ma pojemność równą $N = 0,75 * n + 1$, gdzie n to liczba elementów do zapisania,
     - pod każdym indeksem tablicy znajduje się jednokierunkowa lista SList,
     - w wypadku kolizji w funkcji skrótu, elementy o tym samym indeksie są dopisywane do listy,
     - funkcja skrótu - zamienia łańcuch znakowy klucza na kod ASCII, jednocześnie wykonując działania arytmetyczne (w tym dzielenie modulo przez pojemność tablicy) z dobranymi wartościami zmiennej a i stałej b (zapewnia to badziej równomierny rozkład).
   - klasy Stopwatch i AdvancedStopwatch, wykonujące pomiar czasów korzystając z funkcji gettimeofday().
   - funkcja główna, zarządzająca kolejnością wykonywania zadań.

2. Zapis do tablicy polegał na odczytaniu z pliku klucza (łańcucha znakowego) i wartości mu odpowiadającej. Plik zawierał kilkadziesiąt prawdziwych imion-kluczy, reszta kluczy była generowana losowo (długości kluczy od 4 do 20 znaków). Wartości pod kluczami również były generowane losowo w zakresie od 7 do $10^8 + 7$.

3. Zapis i odczyt elementów tablicy dokonywany był za pomocą przeciążenia indeksującego [ ], działającego podobnie jak przy tradycyjnej tablicy (tzn. możliwa jest operacja tablica["klucz"] = wartość).

4. Pomiary czasów zostały wykonane dla nieco zmodyfikowanych rozmiarów problemu:
$$n = 10, 10^2, 10^3, 10^4, 10^5, 10^6; \tag{2}$$

5. Odczyt z tablicy polegał na wylosowaniu indeksu z pomocniczej tablicy 10-elementowej, zawierającej przykładowe klucze do znalezienia. Były tam również elementy nieistniejące w tablicy z haszowaniem, więc zachodził najgorszy przypadek, kiedy należy przeszukać całą listę pod wskazanym przez funkcję skrótu indeksem. Odczytane pod wylosowanym kluczem wartości były zapisywane do pliku 'Odczytane'.

6. Pomiary wykonano w seriach po 25 razy dla każdego n.

3 Pomiary średnich czasów

1. Zapis elementów do tablicy z haszowaniem:

n	10	10^2	10^3	10^4	10^5	10^6
czas [s]	0.000007	0.000056	0.000701	0.007337	0.082544	0.859796

2. Odczyt pojedynczego elementu:

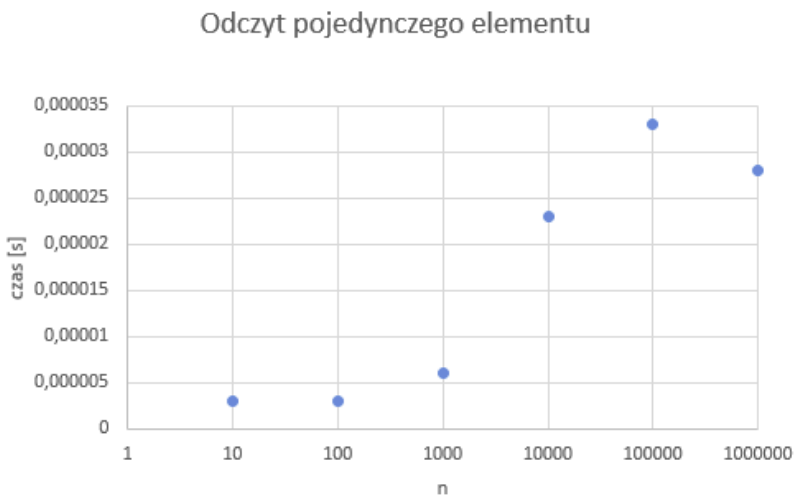
n	10	10^2	10^3	10^4	10^5	10^6
czas [s]	0.000003	0.000003	0.000006	0.000023	0.000033	0.000028

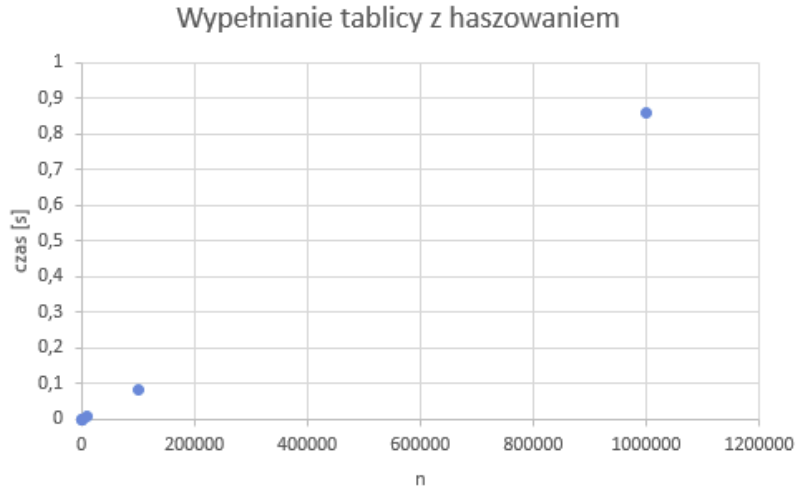
4 Wnioski

Wstawianie elementu do tablicy jest wykonywane w czasie stałym $O(1)$. Dla kolejnych wartości n czas wypełniania tablicy rośnie niemal liniowo, co potwierdza, że operacja przypisania nowej wartości do jeszcze nieistniejącego klucza ma stałą złożoność czasową, a czas wypełnienia całej tablicy zależy bezpośrednio od n.

W przypadku odczytu pojedynczego elementu, wyniki wyraźnie nie są stałe, ale nie rosną też liniowo wraz z n. Ogranicza je długość najdłuższej listy w komórce tablicy, którą przyjdzie przeszukiwać, aby odczytać wartosc pod zadanym kluczem. Długość tej listy nie wynosi n, ale znacznie mniej, dzięki działaniom funkcji skrótu, która rozprasza klucze po całej tablicy w miarę równomiernie. Ostatecznie można powiedzieć, że odczyt (wyszukanie) wartości pod podanym kluczem ma złożoność $O(k)$, gdzie k jest długością najdłuższej listy w tablicy, którą trzeba przeszukać.

5 Wykresy


Odczyt pojedynczego elementu
0,000035
0,00003
0,000025
0,00002
0,000015
0,00001
0,000005
0
czas [s]
1
10
100
1000
10000
100000
1000000
n


Wypełnianie tablicy z haszowaniem
1
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
czas [s]
0
200000
400000
600000
800000
1000000
1200000
n