

Sprawozdanie LAB5

Arkadiusz Ziółkowski

17.04.2015r

1 Cel ćwiczenia

Celem ćwiczenia jest zbadanie złożoności obliczeniowej operacji odczytu/zapisu do/z zaimplementowanej Tablicy Haszującej.

2 Wywód teoretyczny złożoności obliczeniowej

2.1 Zapis

- Zapis polega na haszowaniu klucza i umieszczeniu danej na końcu listy znajdującej się na odpowiednim indeksie tablicy (Array).
- Funkcja haszująca na podstawie wartości liczbowych kodu ASCII zawsze przemnaża wartości sześciu pierwszych liter klucza (pomnożonych przez odpowiedni współczynnik). Wobec czego wykonuje się w stałym czasie $T(1)$ uzyskując złożoność obliczeniową $O(1)$.
- Zgodnie ze Sprawozdaniem LAB3 zapis n elementów do listy (ListArr2x) ma złożoność $O(n)$, więc średni zapis pojedynczego elementu ma złożoność $O(1)$ (czas wykonania $T(1)$).
- Jest oczywiste, że dostęp do k -tego pola Arraya charakteryzuje się czasem $T(1)$ z czego wynika złożoność obliczeniowa $O(1)$.
- W związku z powyższymi czas operacji zapisu do Tablicy Haszującej wyraża się wzorem:

$$\begin{aligned} T_{\text{haszowania}} + T_{\text{dostęp_Array}} + T_{\text{zapis_lista}} &= \\ &= T(1) + T(1) + T(1) = \\ &= T(1) \Rightarrow O(1) \end{aligned}$$

- Złożoność obliczeniowa algorytmu zapisu do Tablicy Haszującej wyraża się w $O(1)$.

2.2 Odczyt

1. Przypadek pesymistyczny

- Zakładamy, że wszystkie n elementów trafiają na tą samą pozycję Array'a Listy Haszującej.
- Szukany element znajduje się na końcu listy podlinkowanej pod daną pozycję.
- Zostanie przeszukana cała lista w celu znalezienia pasującego klucza - n porównań co daje złożoność czasową $T(n)$.
- Zaniedbując pomijalnie małą złożoność funkcji haszującej, oraz dostępu do pola tablicy Array, powyższa złożoność czasowa $T(n)$ implikuje złożoność obliczeniową wyrażoną w $O(n)$.

2. Przypadek średni

- Zakładamy, że prawdopodobieństwo pojawienia się elementów na każdym z idensków tablicy haszującej jest takie samo.
- Przyjmujemy ilość elementów tablicy n mniejszą lub równą jej rozmiarowi.
- Przy jednakowych długościach klucza.
- Odczyt danej sprowadza się do wygenerowania przez funkcję haszującą indeksu tablicy, uzyskania dostępu do pola tablicy Array i porównania zgodności klucza.
- Wobec powyższego czas odczytu:

$$\begin{aligned}T_{haszowania} + T_{dostęp_Array} + T_{porównanie_klucza} &= \\&= T(1) + T(1) + T(1) = \\&= T(1) \Rightarrow O(1)\end{aligned}$$

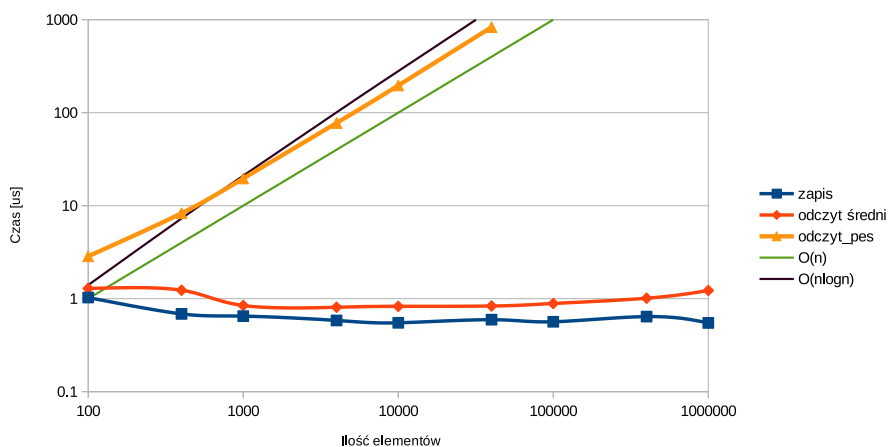
- Średnia złożoność obliczeniowa algorytmu odczytu do Tablicy Haszującej wyraża się w $O(1)$.

3 Wyniki pomiarów

Pomiary zostały wykonane dla średniego i pesymistycznego przypadku odczytu danych. Oraz dla ogólnego przypadku ich zapisu do do Tabli Haszującej. Wynikami pomiarów są średnie czasy zapisu/odczytu pojedynczego elementu z próby.

Ilość elementów	Średni czas obliczeń [us]		
	Zapis	Odczyt średni	Odczyt pesymistyczny
10^2	1,028	1,286	2,853
$4 * 10^2$	0,686	1,2322	8,299
10^3	0,649	0,844	19,625
$4 * 10^3$	0,583	0,808	77,507
10^4	0,550	0,826	196,346
$4 * 10^4$	0,593	0,833	831,965
10^5	0,564	0,886	-
$4 * 10^5$	0,641	1,010	-
10^6	0,548	1,224	-

Tabela 1. Wyniki pomiarów odczytu/zapisu



Rysunek 1. Wykres czasu od ilości danych

4 Wnioski

- Z Rysunku 1. bardzo łatwo można zauważyć, iż odczyt pesymistyczny uzyskany drogą ekperymentalną posiada złożoność $O(n)$, odczyt średni $O(1)$ oraz zapis $O(1)$.
- Wyniki pomiarów przedstawione na Rysunku 1. i w Tabeli 1. są zgodne z wcześniejszymi wywodami teoretycznymi.
- Tablica Haszująca będąca szczególnym przypadkiem Tablicy Asocjacyjnej jest doskonałą implementacją kontenera przechowującego dane pod kluczem ze względu na szybki zapis oraz i średni odczyt - $O(1)$.
- Podczas implementacji należy szczególną uwagę poświęcić funkcji haszującej, która powinna zwracać wartości liczbowe (z zakresu rozmiaru tablicy) z rozkładem prawdopodobieństwa jak najbardziej zbliżonym do równomiernego.