Laboratorium Projektowania algorytmów i metod sztucznej inteligencji

Tablica asocjacyjna, haszowanie

218582

18 kwietnia 2016

Spis treści

Spis rysunków

1 Wstęp

Celem zadania było stworzenie tablicy asocjacyjnej z mechanizmem haszowania.

Utworzona tablica asocjacyjna zawiera w sobie tablicę elementów zwanych koszykami (bucket), z których każdy posiada tablicę wpisów (entry) zawierających klucz (key) oraz wartość (value). Dodanie wpisu klucz-wartość do tablicy asocjacyjnej powoduje dodanie wpisu do wyliczonego w funkcji haszującej odpowiedniego koszyka, na kolejne miejsce w tablicy koszyka. Wyszukanie elementu po kluczu powoduje obliczenie, w którym koszyku znajduje się element (o ile istnieje), a następnie przeszukanie tablicy elementów tylko w tym koszyku w poszukiwaniu wpisu o zadanym kluczu.

Zapis jednego elementu do tablicy asocjacyjnej jak i wyszukanie pojedynczego elementu powinny charakteryzować się złożonością $O(1)$.

2 Użyte rozwiązania

Funkcja haszująca korzysta z `std::accumulate`, która zlicza wartości kodów ASCII poszczególnych znaków klucza. Następnie ta wartość jest poddawana działaniu modulo ilość koszyków.

Ilość koszyków jest zwiększana w zależności od wielkości testu. Test 100 i mniej elementów powoduje utworzenie 3 koszyków. Dla większej ilości, koszyków jest 33-krotnie mniej niż elementów (dzielenie całkowitoliczbowe).

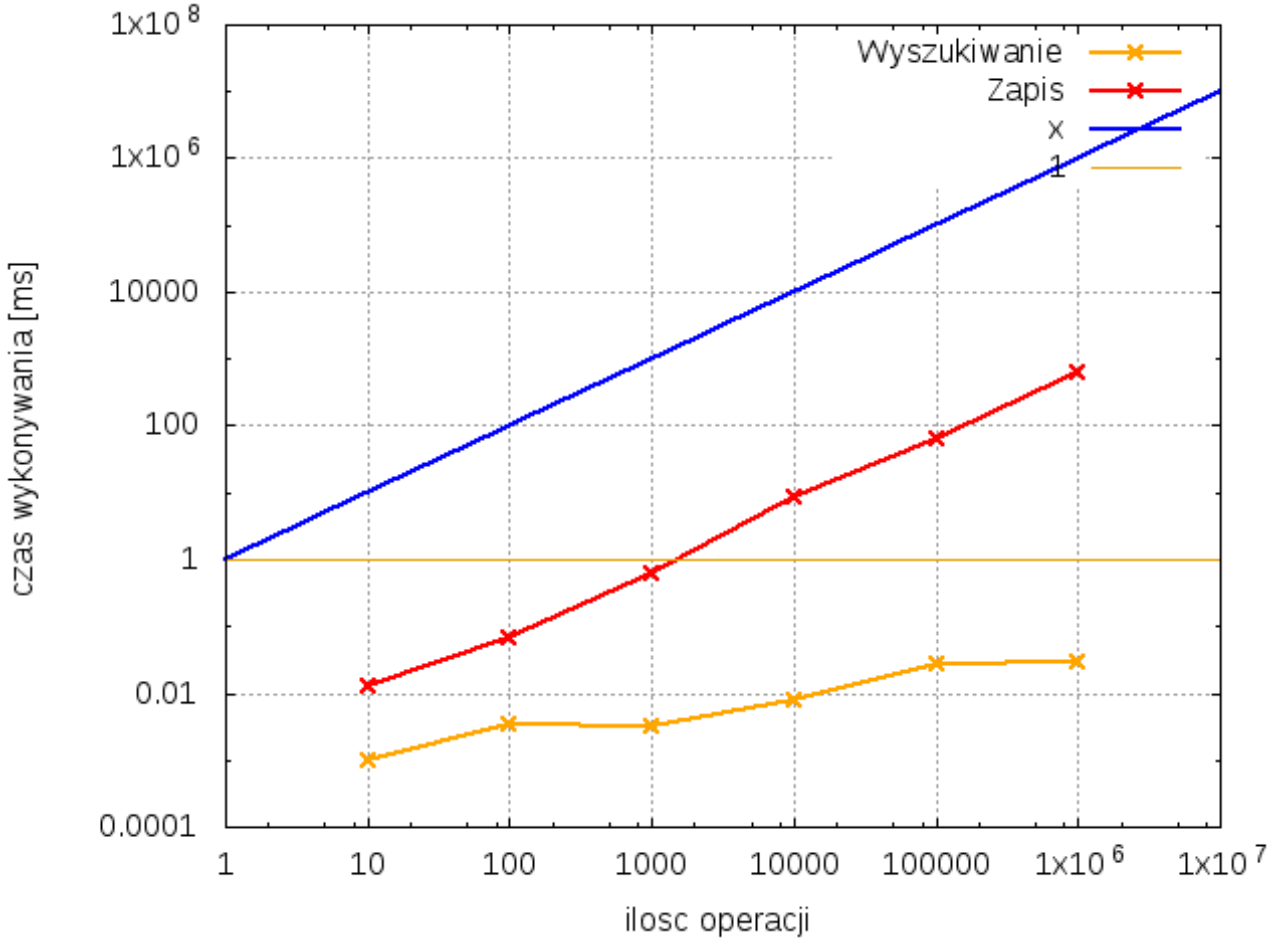
Do koszyków wpisywane są elementy typu `entry`. Jest to klasa posiadająca pola `key, value`, oraz odpowiednie metody.

Utworzono interfejsy, przeciążono niezbędne operatory klasy `entry`, `hash` (koszyk) i `Asoc` (tablica asocjacyjna) oraz dodano potrzebne przeciążenia tablicy dynamicznie rozszerzalnej.

Ponieważ znamy wcześniej wielkość problemu, umożliwia to uniknięcie ponownego haszowania wszystkich danych, poprzez odpowiednie przygotowanie funkcji haszującej - uzależnienie jej od ilości danych wejściowych.

3 Wyniki

Przy użyciu wyżej wymienionych rozwiązań zebrano dane dotyczace czasu wykonywania się programu, zależnie od wielkości zadanego problemu: dla 10, 100, 1000, 10000, 100000 i 1000000 elementów. Wyniki przedstawia wykres na Rys. 1.



Rysunek 1: Czas wczytywania danych do tablicy i przeszukiwania

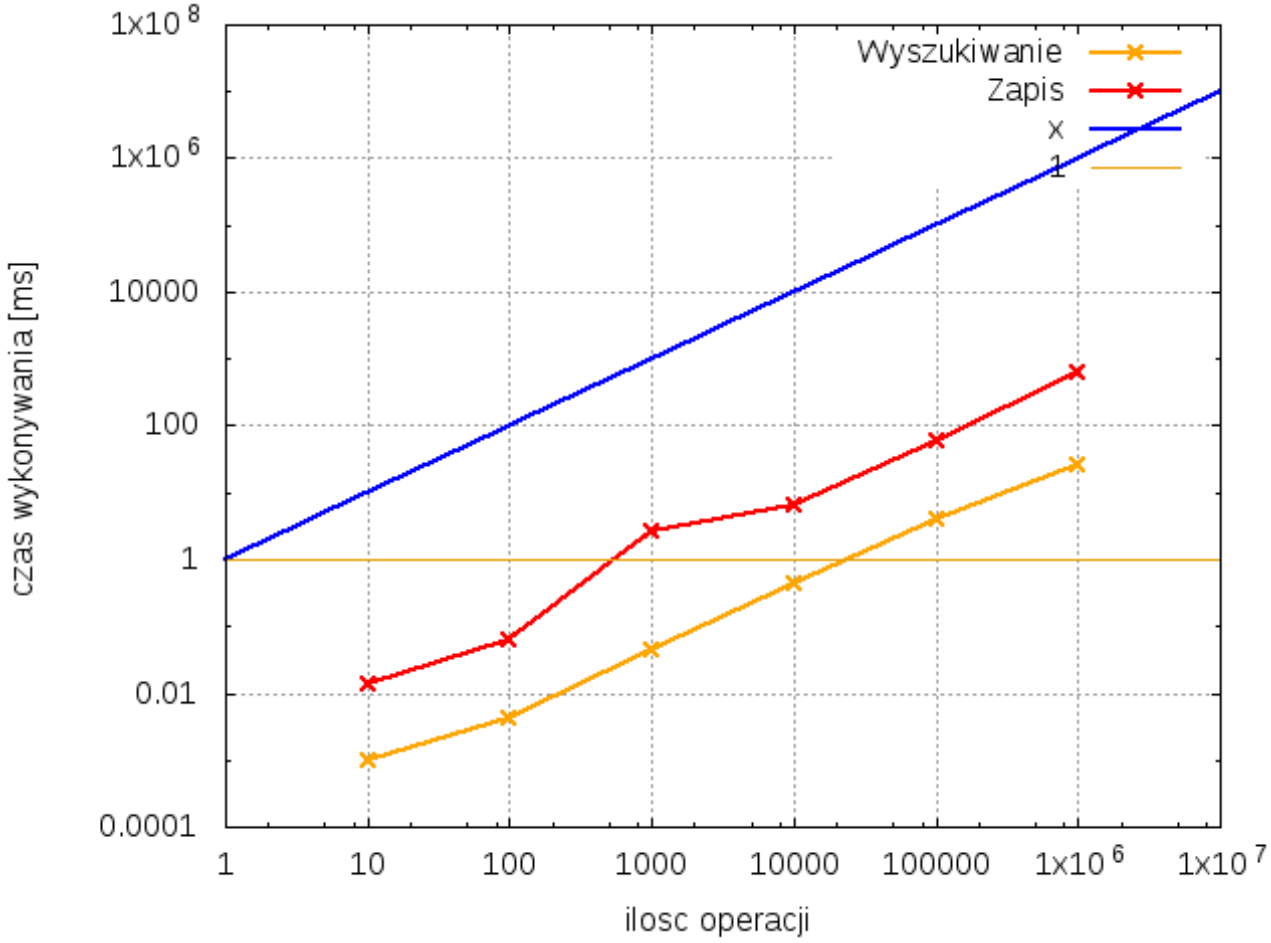
Można zauważyć na wykresie, że złożoność empiryczna zapisu elementów do tablicy asocjacyjnej pokrywa się z teoretyczną - zapis n elementów o złożoności $O(1)$ daje złożoność całego procesu: $O(n)$.

Jednakże wyraźnie można zauważyć, że przeszukiwanie tablicy asocjacyjnej odbiega od złożoności $O(1)$. Wynika to prawdopodobnie z faktu, że w rozwiązaniu nie wykorzystuje się stałej maksymalnej wielkości koszyków. Teoretycznie losowe wpisy powinny rozłożyć się równo po każdym koszyku dla zwiększającej się ilości elementów, ale oczywiście może tak nie być.

Niezachowanie złożoności $O(1)$ przy odczycie może mieć swoje źródło również w doborze ilości koszyków do wielkości problemu. Obecne rozwiązanie jest dobrane zupełnie przypadkowo.

Istotnie, dalsze dostosowywanie ilości koszyków do wielkości problemu poprawiło czas odczytu, ale nie na tyle, by na całym zakresie uzyskać złożoność $O(n)$.

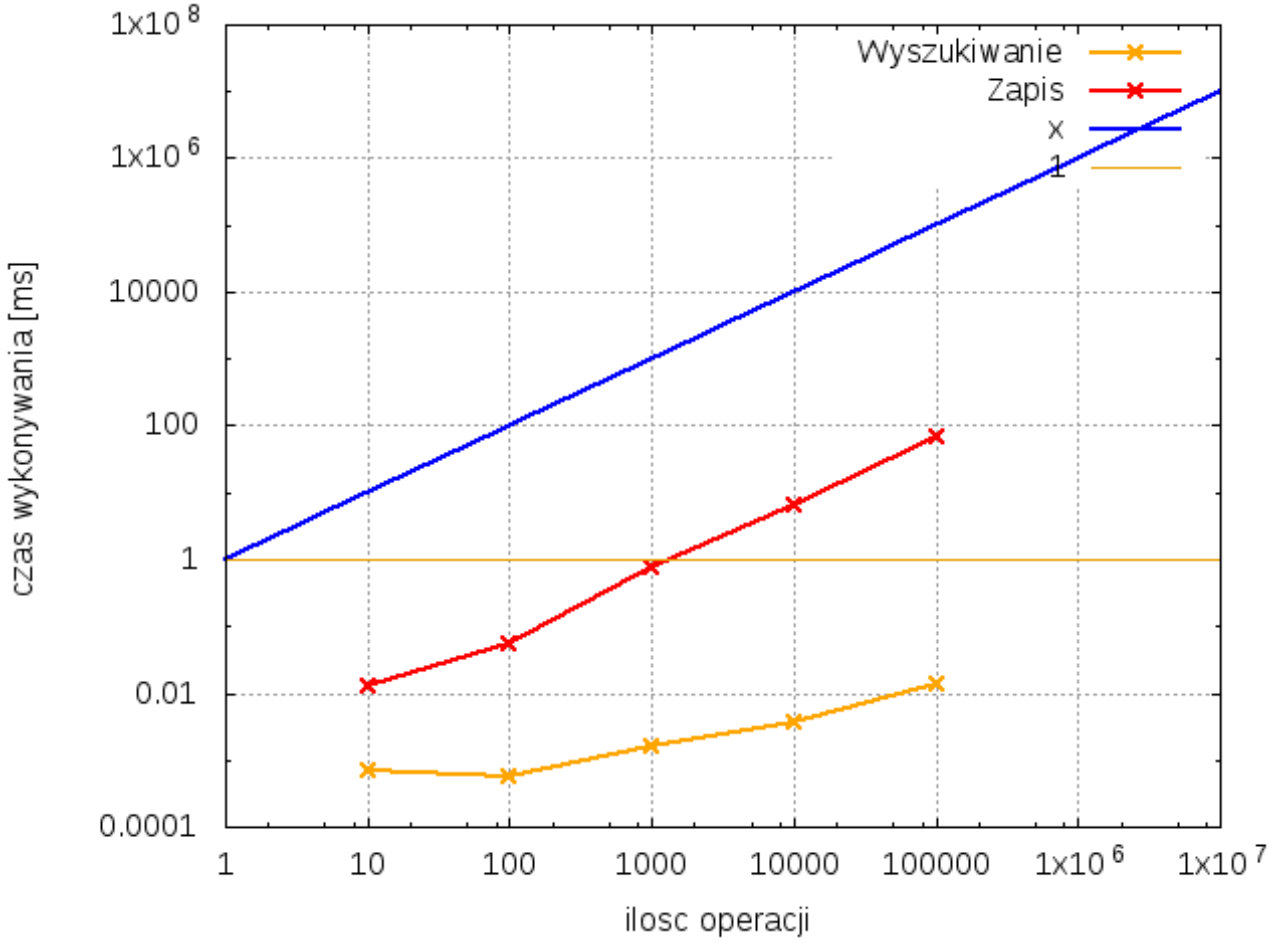
Sprawdzono również, w jaki sposób zmieni się czas odczytu, gdy będzie tylko jeden koszyk. Wyniki prezentuje wykres na rys. 2.



Rysunek 2: Czas wczytywania danych do tablicy i przeszukiwania dla jednego koszyka

W tym przypadku złożoność wyszukiwania wzrosła do $O(n)$. Jest to przypadek pesymistyczny i tutaj wynik empiryczny pokrywa się z teoretycznym.

Ponadto wykonano test dla sytuacji, w której zostanie stworzone tyle koszyków co wpisów. Wyniki prezentuje wykres na rys. 3.



Rysunek 3: Czas wczytywania danych do tablicy i przeszukiwania dla ilości koszyków równej ilości wpisów

Dla miliona elementów nastąpił problem z brakiem pamięci operacyjnej i nie udało się przeprowadzić testu. Może to wynikać z implementacji tablicy dynamicznie rozszerzalnej, która za każdym razem na początku swojego istnienia alokuje pamięć dla 10 elementów.

4 Wnioski

O ile złożoność obliczeniowa zapisu nie jest trudna do uzyskania, gdyż jest złożonością zapisu elementów do tablicy dynamicznej, o tyle złożoność teoretyczna $O(1)$ dla wyszukania elementu z tablicy asocjacyjnej nie jest zadaniem trywialnym. Należy dobrać ilość koszyków tak, aby wypośrodkować między dużą złożonością czasową (jeden koszyk) a dużą złożonością pamięciową (tyle koszyków ile wpisów).

Aby poprawić działanie tablicy asocjacyjnej, należałoby skupić się zarówno na dobraniu optymalnego doboru ilości koszyków oraz zbadaniu innych funkcji haszujących.