

# Sprawozdanie z ćwiczeń laboratoryjnych nr 8

## Tablice z haszowaniem

Kacper Kafara

nr grupy 9

śr. 14:40

14 maja 2020

# 1 Zadanie

Zgodnie z "Instrukcją do ćwiczeń laboratoryjnych":

- Porównanie czasów wykonania oraz liczby *kolizji* w algorytmach haszowania otwartego dla różnych danych wejściowych.
- Pomiar czasu wykonania algorytmu haszowania łańcuchowego dla różnych danych wejściowych.

# 2 Metodologia

- Do pomiaru czasu wykonania algorytmów wykorzystana została funkcja `clock_t clock()`.
- Pomiar czasu wykonania algorytmów opierał się na:
  - wygenerowaniu bazy danych (o zadanym rozmiarze  $n$ ) złożonej z rekordów  $[name, number]$
  - wygenerowaniu sekwencji  $k$  zapytań (add / remove / get) oraz numerów rekordów do których odnosi się poszczególne zapytanie
  - 100-krotnym pomiarze czasu wykonania wszystkich 3 algorytmów (każdy pomiar dla tej samej bazy danych, ale za każdym razem generowana nowa  $k$  elementowa sekwencja zapytań)
  - za czas wykonania algorytmu przyjęta została średnia arytmetyczna ze wszystkich 100 prób.
- Liczba kolizji to podobnie  $\lfloor srednia - arytmetyczna - z - liczby - kolizji \rfloor$ .

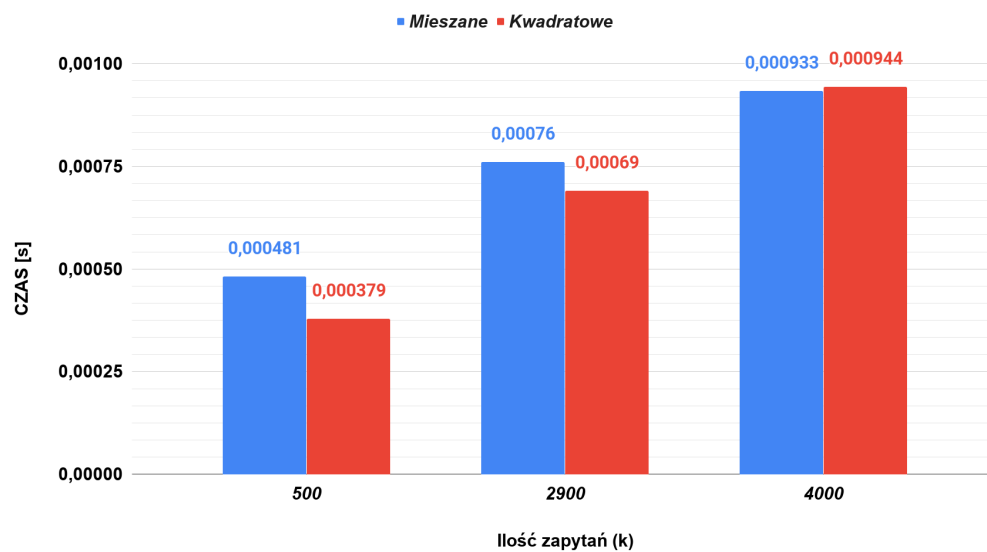
# 3 Rezultaty

W dalszej części sprawozdania zamieszczona została seria wykresów czasu wykonania  $t$  ( $[t] = s$ ) poszczególnych algorytmów haszowania w zależności od liczby zapytań przy ustalonej wielkości bazy danych oraz - w przypadku haszowania otwartego - także wykresy ilości kolizji od ilości zapytań. W lewym górnym rogu każdego z wykresów znajdziemy rozmiar bazy danych  $n$  dla którego były wykonywane pomiary. Na osi poziomej liczba zapytań  $k$ . Dla każdego  $n$  zostały wybrane 3 liczby zapytań: znacznie mniejsza od rozmiaru bazy, nieco mniejsza niż rozmiar bazy oraz przewyższająca rozmiar bazy.

## 3.1 Haszowanie otwarte

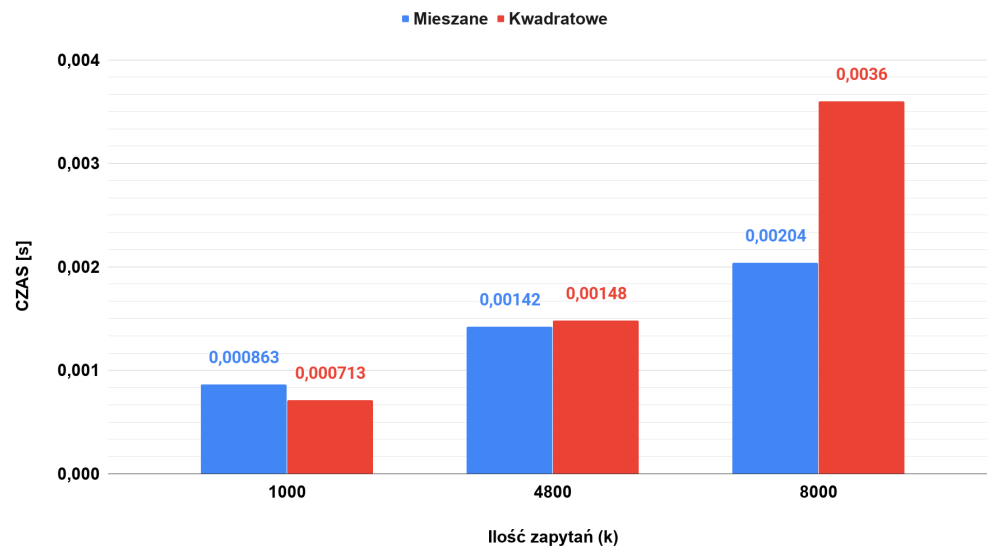
Kolorami oznaczone rodzaje haszowania otwartego.

Wielkość bazy  $n == 3000$



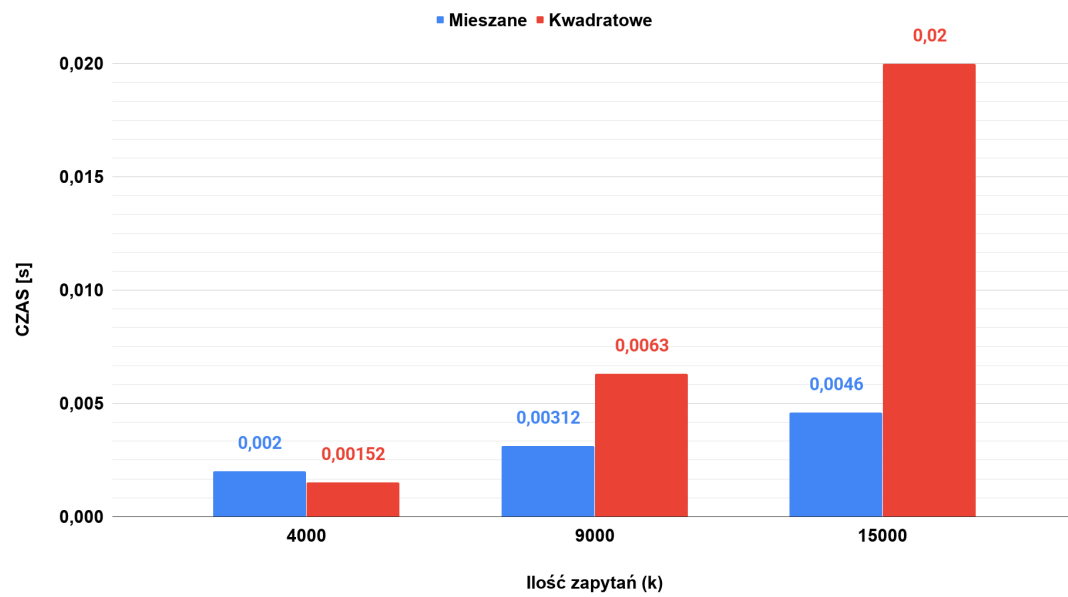
Rysunek 1:  $\text{czas}(\text{ilość\_zapytań})$ ,  $n = 3000$

Wielkość bazy  $n == 5000$



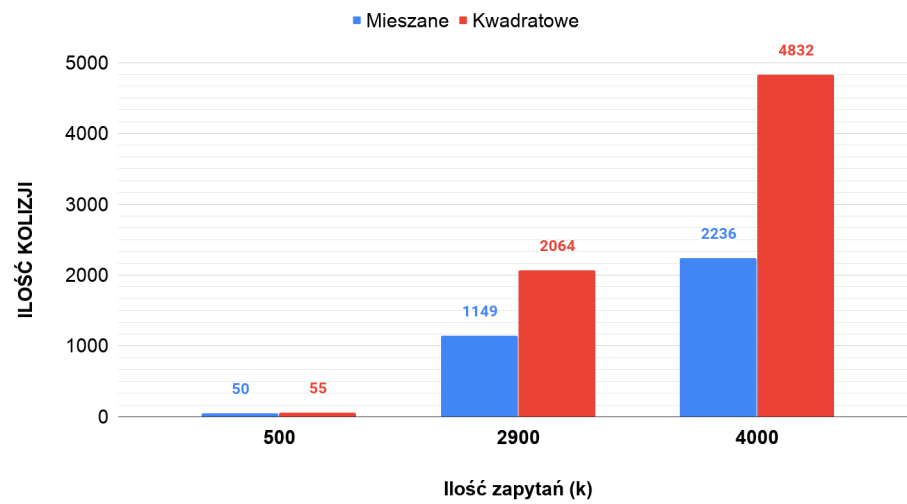
Rysunek 2:  $\text{czas}(\text{ilość\_zapytań})$ ,  $n = 5000$

Wielkość bazy  $n = 10000$



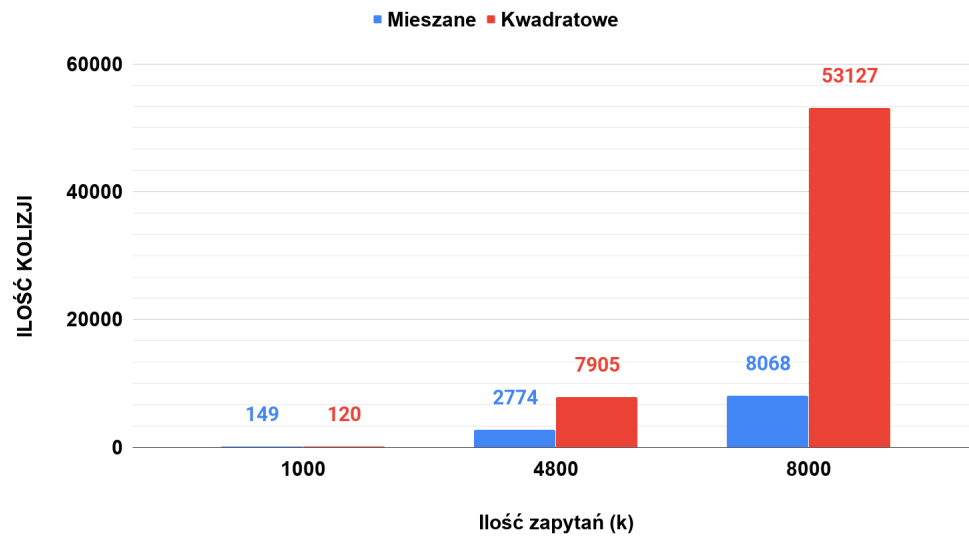
Rysunek 3:  $\text{czas}(\text{ilość\_zapytań})$ ,  $n = 10000$

Wielkość bazy  $n = 3000$



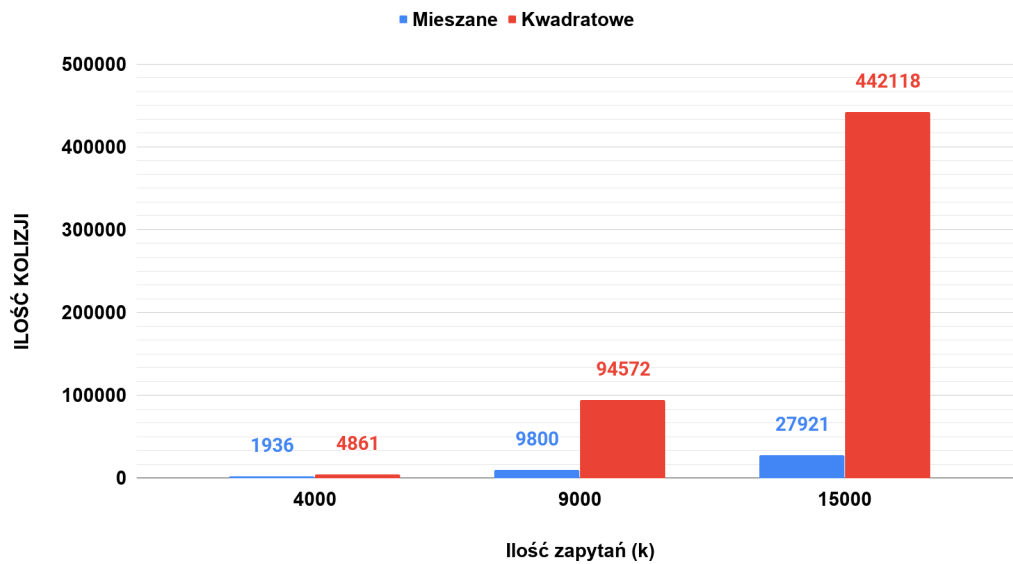
Rysunek 4:  $\text{ilość\_kolizji}(\text{ilość\_zapytań})$ ,  $n = 3000$

Wielkość bazy  $n = 5000$



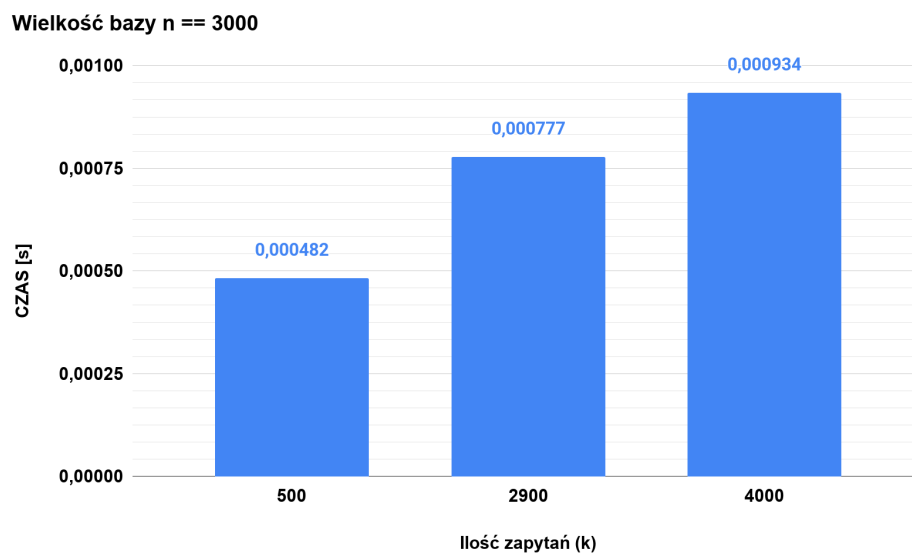
Rysunek 5: ilość\_kolizji(ilość\_zapytań),  $n = 5000$

Wielkość bazy  $n = 10000$

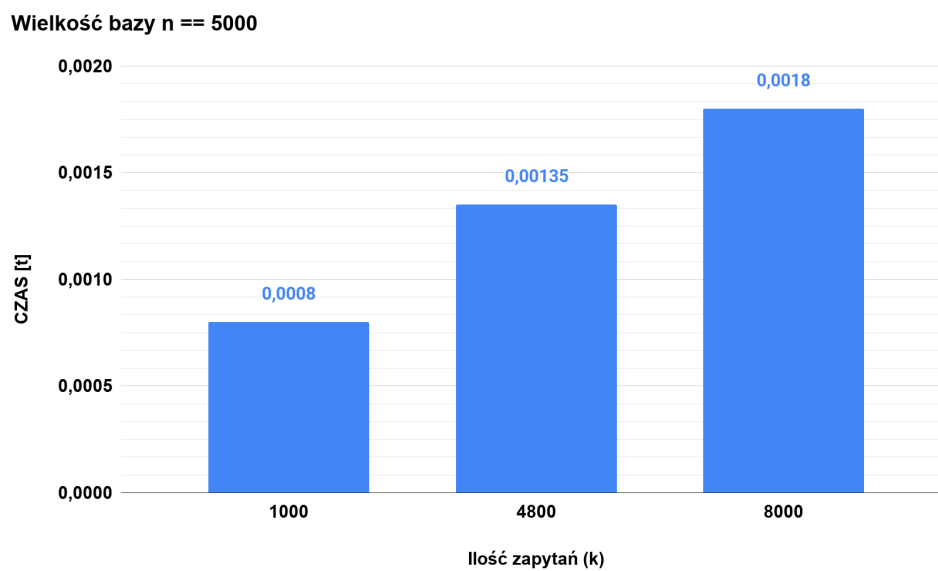


Rysunek 6: ilość\_kolizji(ilość\_zapytań),  $n = 10000$

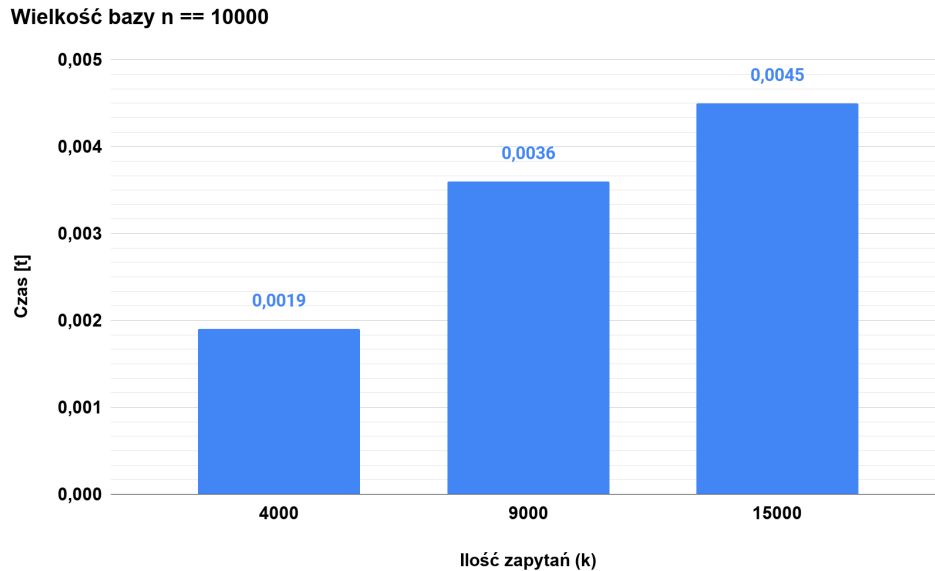
### 3.2 Haszowanie łańcuchowe



Rysunek 7:  $\text{czas}(\text{ilość\_zapytań})$ ,  $n = 3000$



Rysunek 8:  $\text{czas}(\text{ilość\_zapytań})$ ,  $n = 5000$



Rysunek 9:  $\text{czas}(\text{ilość\_zapytań})$ ,  $n = 10000$

## 4 Wnioski

### 4.1 Haszowanie otwarte

- Metoda z adresowaniem kwadratowym wydaje się być szybsza dla stosunkowo małych baz danych oraz liczby zapytań niższej niż mocy całego uniwersum kluczy. Rzadko jednak wiemy jak liczny jest zbiór wszystkich możliwych kluczy  $\Rightarrow$  metoda z adresowaniem mieszanym obciążona jest mniejszym ryzykiem gwałtownego wzrostu czasu wykonania.
- Adresowanie mieszane daje znacznie bardziej stabilne rezultaty niż adresowanie kwadratowe.
- Adresowanie mieszane prowadzi do znacznie mniejszej ilości kolizji (a w takim razie mamy mniej tracenia czasu na dodatkowe obliczenia), szczególnie gdy ilość zapytań zbliża się do mocy uniwersum kluczy lub go przewyższa.
- Widzimy, że adresowanie kwadratowe w pewnych przypadkach z pewnością nie działa w czasie stałym. (Można popatrzeć na bardzo zmienny stosunek czasu wykonania algorytmu od ilości zapytań, szczególnie dla  $k \gg n$ ).

### 4.2 Haszowanie łańcuchowe

- Dla losowych (pseudolosowych) danych daje czasy zbliżone do adresowania mieszanego.
- Jeżeli popatrzymy na stosunek czasu wykonania algorytmu do liczby zapytań to możemy zauważyć że jest on w przybliżeniu stały ( $\simeq 0,24 * 10^{-6}$ ) dla różnych  $n$  i  $k$ , co odróżnia ten algorytm od algorytmu z adresowaniem kwadratowym.

- Z poprzedniego punktu wynika, że algorytm działa w spodziewanym czasie stałym.