

Sprawozdanie z ćwiczeń laboratoryjnych nr 9

Skip listy

Kacper Kafara
nr grupy 9
śr. 14:40

27 maja 2020

1 Zadanie

Pomiar czasu wykonania operacji:

- wstawiania elementu do skip listy
- usuwania elementu z skip listy
- wyszukiwania elementu w skip liście

oraz porównanie tych czasów dla 2. różniących się implementacji skip listy.

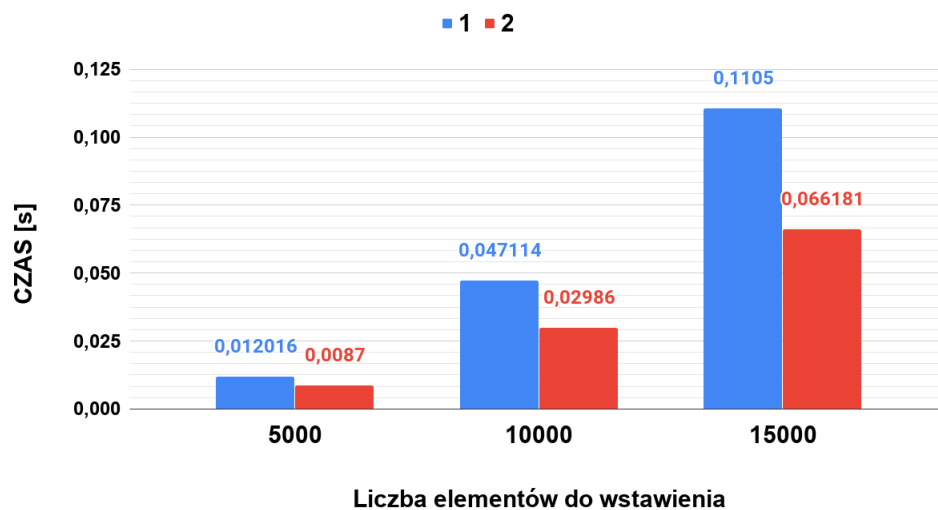
2 Metodologia

- Do pomiaru czasu wykonania algorytmów wykorzystana została funkcja *clock_t clock()*.
- Obydwie implementacje były testowane na tych samych danych.
- Za czas wykonania operacji przyjęty został czas średni (dla każdego danych implementacje były testowane wielokrotnie)

3 Rezultaty

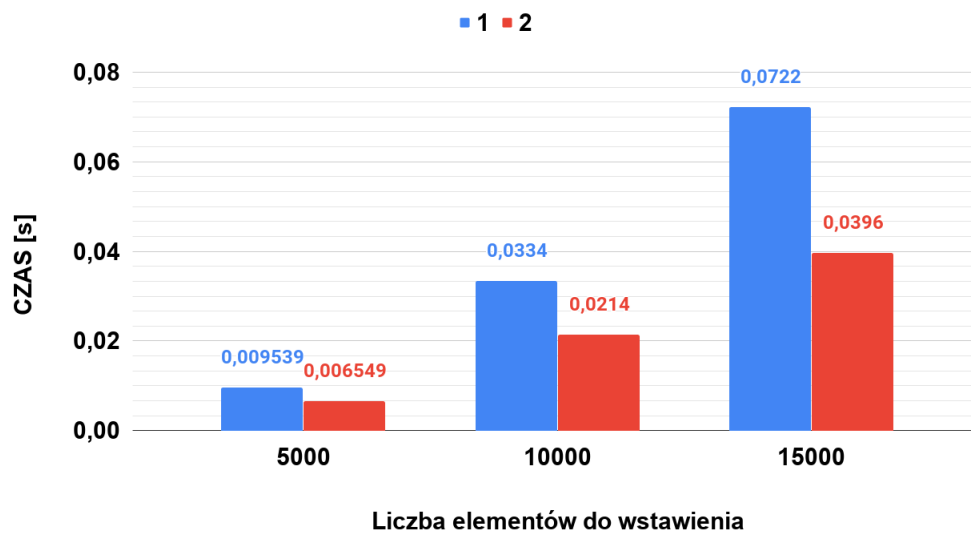
W dalszej części zamieszczona została seria wykresów czasu wykonania w zależności od charakterystyki danych wejściowych. Na wykresach przez skrót *maxh* rozumiana jest maksymalna wysokość węzłów skip listy. Przez (1) została oznaczona implementacja skip listy bez zachowania zasady lokalności odwołań. (2) to implementacja z zachowaniem lokalności odwołań.

Operacja insert, maxh == 10



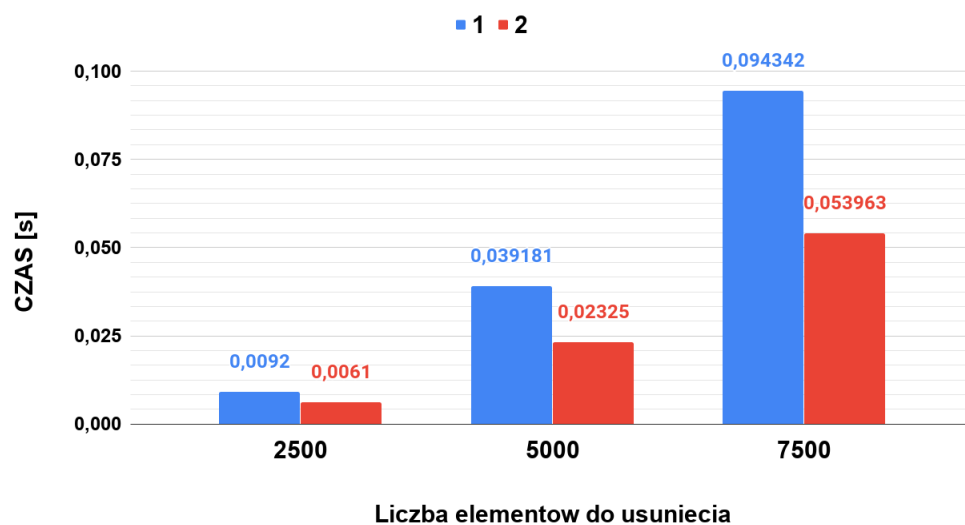
Rysunek 1: $maxh = 10$

Operacja insert, maxh == 15



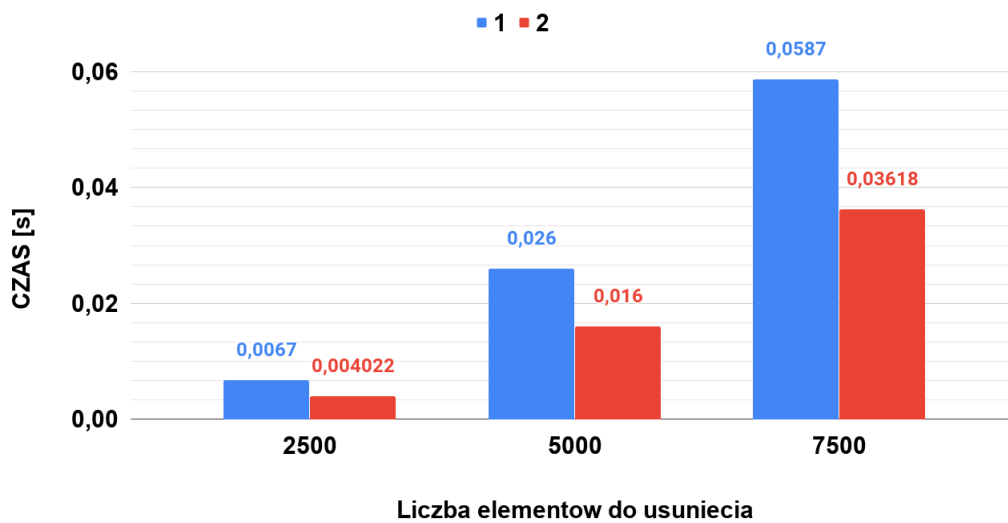
Rysunek 2: $maxh = 15$

Operacja remove, maxh == 10



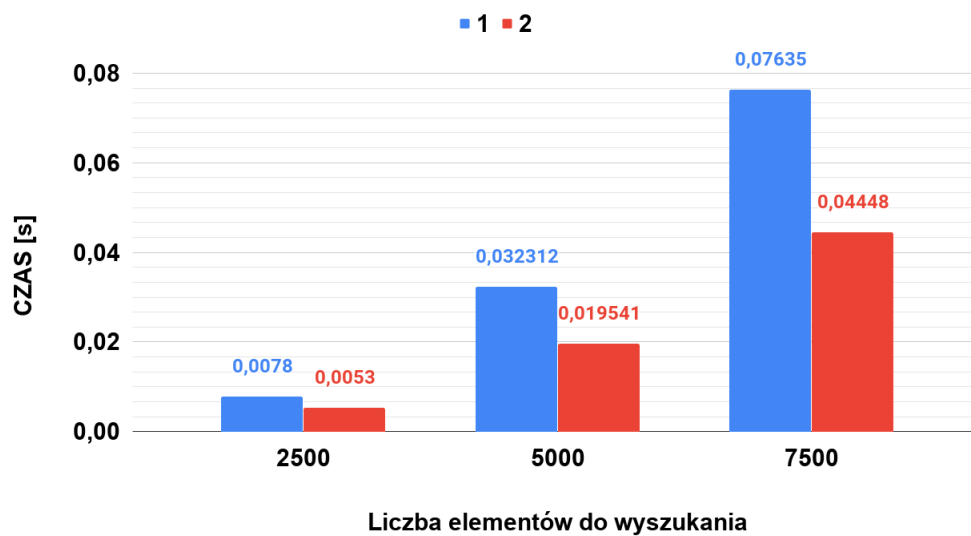
Rysunek 3: $maxh = 10$

Operacja remove, maxh == 15



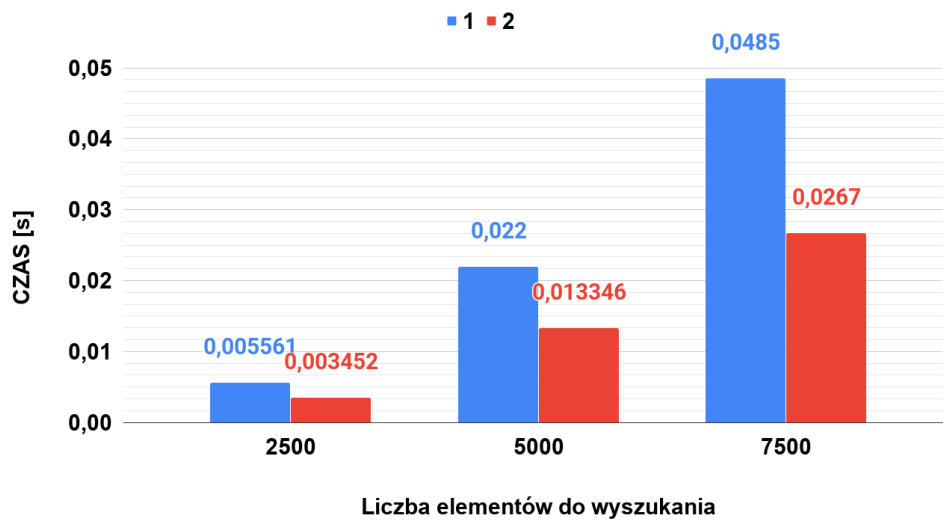
Rysunek 4: $maxh = 15$

Operacja search, maxh == 10



Rysunek 5: $maxh = 10$

Operacja search, maxh == 15



Rysunek 6: $maxh = 10$

Jeżeli popatrzymy na stosunki $K_{operacja}$ czasów wykonania odpowiadających sobie algorytmów z implementacji (1) do (2), to w przybliżeniu, niezależnie od ilości zapytań, dla:

- $maxh = 10$ otrzymamy:

- $K_{insert} \approx 1,52$
- $K_{remove} \approx 1,66$
- $K_{search} \approx 1,63$

- $maxh = 15$ otrzymamy:

- $K_{insert} \approx 1,67$
- $K_{remove} \approx 1,65$
- $K_{search} \approx 1,72$

4 Wniosek

- zastosowanie zasady lokalności odwołań pozwala na przyspieszenie każdej z operacji na skip liście ok. 1,6-krotnie