Sprawozdanie z laboratorium PAMSI

Damian Oleksak

Wstęp i wyniki pomiarów

Do rozwiązywania różnego rodzaju problemów służą odpowiednie algorytmy. Do pracy algorytmu niezbędne są struktury danych, które przechowują informacje. W zależności od charakterystyki rozwiązywanego problemu, stawiamy tym strukturom określone wymagania.

Tablice asocjacyjne przechowują dane parami, na które składają się wartość i klucz. Aby uzyskać dostęp do żądanej wartości, posługujemy się kluczem. Zależy nam na jak najbardziej efektywnym czasie dostępu do wybranej wartości. Czas zapisu mniej nas interesuje i jest zwykle dłuższy od czasu odczytu.

Tablica asocjacyjna została zrealizowana w trzech wersjach:
-tablica przy użyciu std::vector;
-drzewo poszukiwań binarnych;
-tablica mieszająca.

Sprawozdanie zawiera porównanie czasu dostępu do ostatniego elementu.

Tabela 1: Wyniki czasu dostępu do ostatniego elementu tablicy mieszającej

LICZBA ELEMENTÓW	CZAS [us]
10	26
100	21
1000	28
10000	30
20000	36
30000	37
40000	35
50000	36
60000	37

Wykres 1. Tablica mieszająca

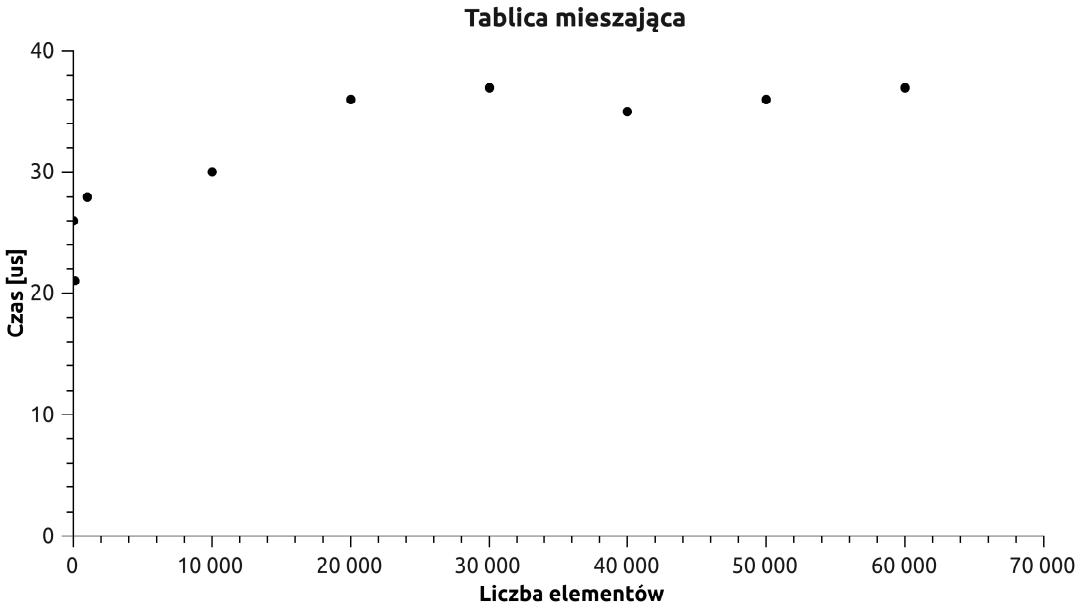


Tabela 2: Wyniki czasu dostępu dla drzewa poszukiwań binarnych

LICZBA ELEMENTÓW	CZAS [us]
10	1
100	2
1000	15
10000	41
20000	88
30000	150
40000	217
50000	276
60000	319

Wykres 2. Drzewo poszukiwań binarnych

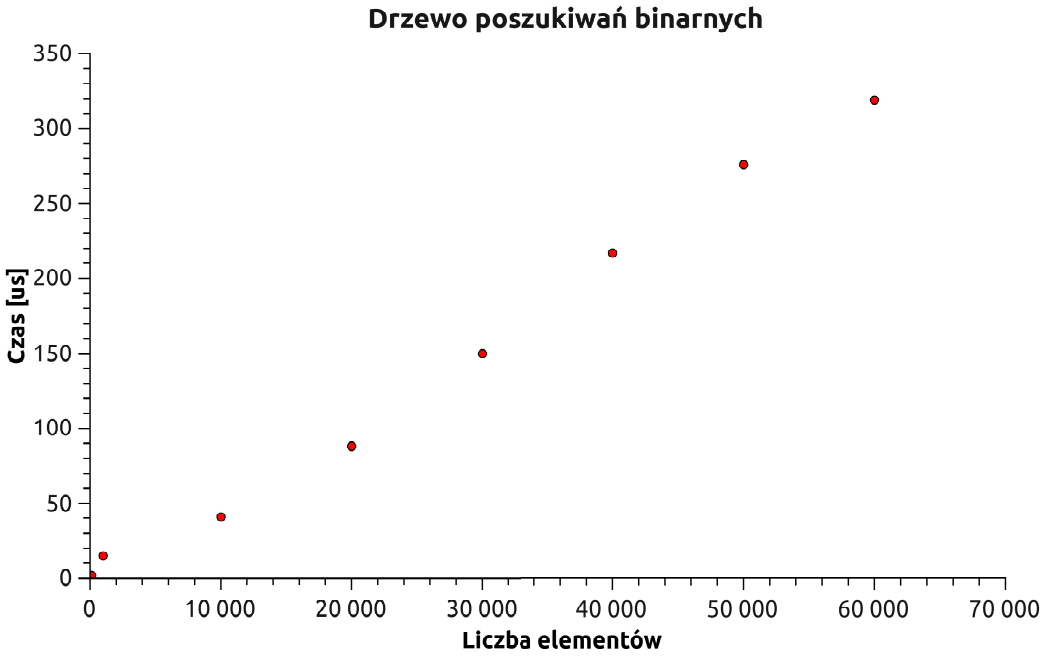
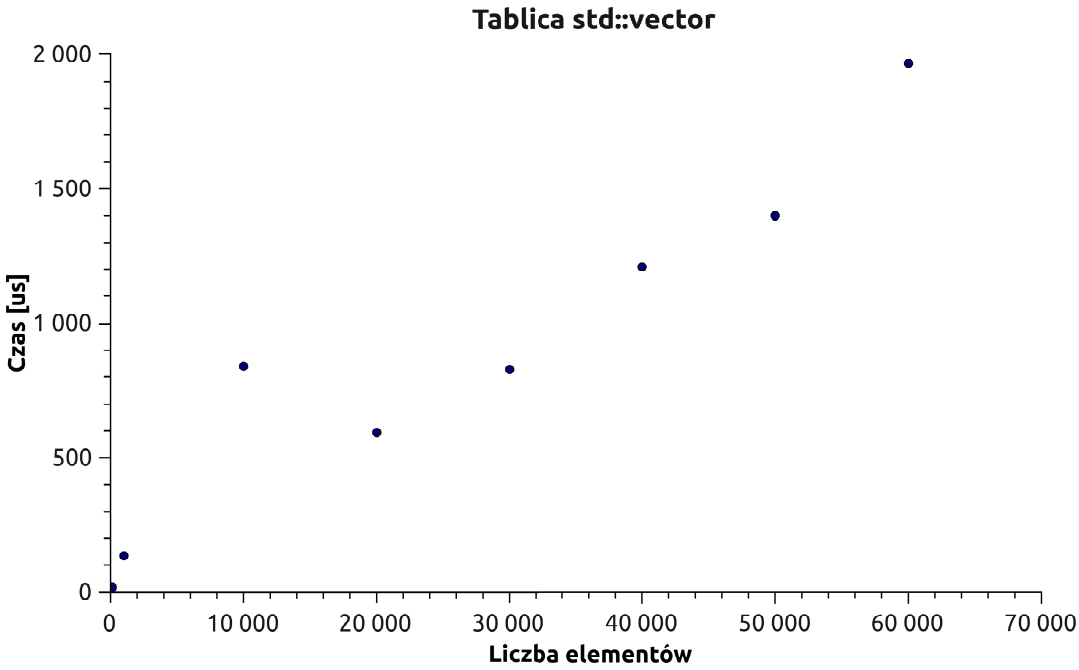


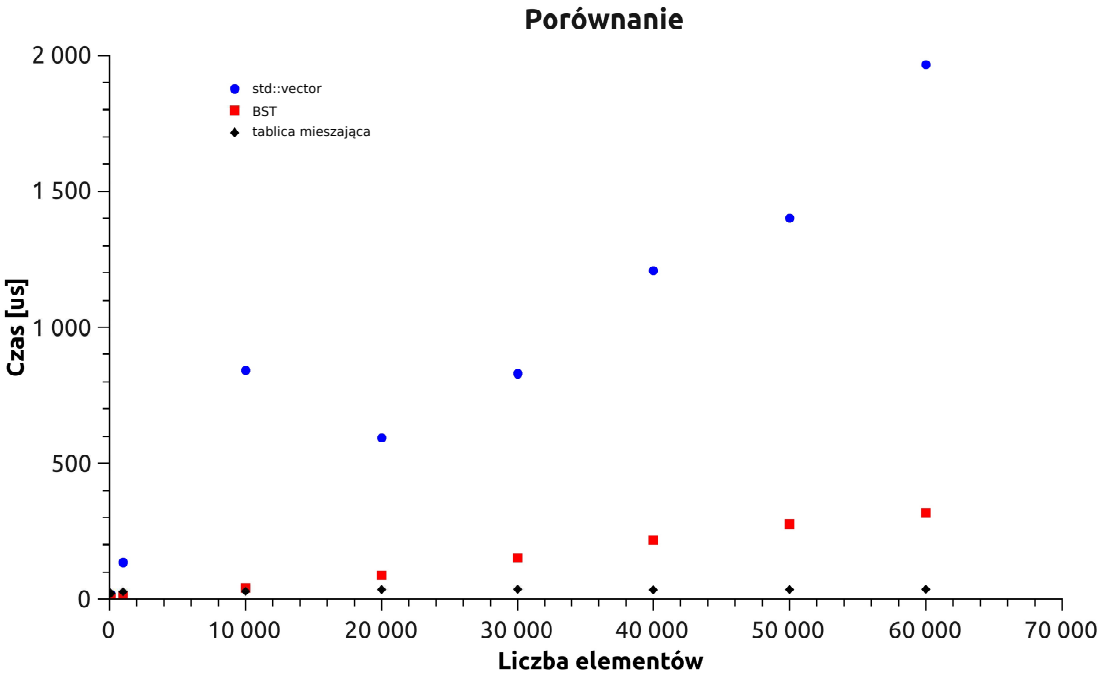
Tabela 3: Wyniki czasu dostępu dla tablicy na std::vector

LICZBA ELEMENTÓW	CZAS [us]
10	16
100	18
1000	136
10000	841
20000	594
30000	829
40000	1209
50000	1400
60000	1966

Wykres 3. Tablica przy użyciu std::vector



Wykres 4. Zbiorcze porównanie



Uwagi i wnioski

Drzewo poszukiwań binarnych (BST) to drzewo, w którym dla każdego węzła "x" musi być spełniony następujący warunek:
Wartość każdego elementu leżącego w lewym poddrzewie węzła x jest nie większa niż wartość węzła x, natomiast wartość każdego elementu leżącego w prawym poddrzewie węzła x jest nie mniejsza niż wartość tego węzła.

Przeszukiwanie odbywa się tylko wzdłuż drzewa, więc czas operacji jest proporcjonalny do jego wysokości. Przypadek optymistyczny wynosi O(logn). Gdy drzewo jest skrajnie niezrównoważone i jego wysokość jest równa liczbie węzłów, koszt operacji wynosi O(n).

Dla tablicy haszującej kluczowym elementem jest jej funkcja haszująca. Zwraca ona dla określonej wartości elementu wartość z zakresu [0,m-1], gdzie m jest rozmiarem tablicy. Funkcja ta musi być deterministyczna, tzn. zawsze dla elementu o wartości k musi zwrócić tą samą wartość.

Ważne jest równomierne haszowanie, by wszystkie listy tablicy haszującej miały podobną długość. Skraca to czas wykonywania operacji słownikowych. (Najkrótszy byłby, gdyby wszystkie listy miały identyczną długość).

Zaimplementowana funkcja mieszająca mnoży wartość znaku(za pomocą przesunięcia bitowego) i sumuje ją z poprzednim wynikiem. Przydzielony hasz jest wynikiem dzielenia modulo tej sumy przez rozmiar tablicy.

Wyniki pomiarów obrazują, że najwydajniejszym rozwiązaniem jest tablica mieszająca. Taki wynik został osiągnięty dzięki odpowiednio dużemu rozmiarowi tej tablicy, przez co elementy rozkładają się równomiernie. Gdyby występowały kolizje, konieczne byłoby przeszukiwanie w listach. Sprawdzanie każdego elementu wydłuża czas liniowo, więc dla bardzo licznych zbiorów czas dostępu znacząco by się wydłużył. Za przykład może posłużyć tablica o rozmiarze 7. Czas odczytu wyniósł 769 us dla 30000 elementów. W porównaniu do 37 us, gdy rozmiar tablicy przekraczał liczbę elementów, różnica jest zauważalna.

Zwykła tablica asocjacyjna bez funkcji mieszającej ma najdłuższe czasy odczytu, gdyż przeszukiwanie odbywa się po każdym elemencie.