

Porównanie odczytu wartości z tablic asocjacyjnych w L^AT_EX-u vol.3

Anna Pływaczyk, 200340

14.06.2014

1 Wstęp

Celem ćwiczenia jest zbadanie złożoności obliczeniowej dostępu do elementów o zadanym kluczu i wartości ze względu na rodzaj dodawanej tablicy: tablica haszująca, asocjacyjna przy użyciu listy i do tablicy stworzonej na podstawie drzewa binarnego. Trzy rodzaje tablic służą do przechowywania abstrakcyjnych typów danych. Stworzone są one przy użyciu szablonu, dzięki któremu do klucza (będącego string’iem) możemy przypisać dowolną wartość (w moim programie od klucza przypisujemy int’y).

Podane wartości pomiaru czasu są liczbami średnimi wykonywanymi na podstawie kilku prób, aby pomiar czasu był bardziej dokładny i zbliżony do prawdziwego.

Pomiar czasu został zmierzony na podstawie różnicy aktualnego czasu systemowego po zakończeniu algorytmu, a czasem przed dostępem do elementu. Podany wynik podany jest w milisekundach.

Pomiary zostały wykonane na systemie Windows, wyniki te mogły zostać przez system sfalszowane poprzez uruchomienie procesów w tle.

2 Rodzaje dostępu do elementów o zadanym kluczu i wartości

2.1 Tablica asocjacyjna przy użyciu listy

Tablica asocjacyjna potocznie zwana również tablicą skojarzeniową. Tablica ta została zaimplementowana na strukturze listy. Najpierw stworzona zostaje struktura wiążąca klucz z obiektem, a następnie stworzony zostaje kontener, w którym są metody działania na elementach o zadanym kluczu i wartości.

Przeszukiwanie tablicy przy użyciu listy jest bardzo powolnym sposobem w porównaniu do gotowego pojemnika asocjacyjnego `std::map`.

2.2 Tablica asocjacyjna przy użyciu drzewa binarnego

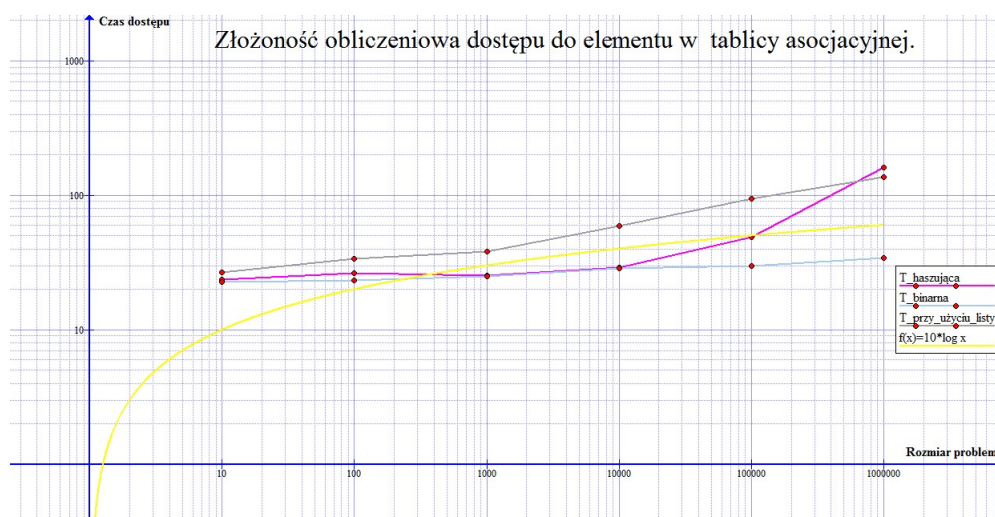
Binarne drzewo poszukiwań jest dynamiczną strukturą, w którym lewe poddrzewo każdego węzła zawiera wyłącznie elementy o kluczach nie większych niż klucz węzła a prawe poddrzewo zawiera wyłącznie elementy o kluczach nie mniejszych niż klucz węzła. Węzły oprócz klucza przechowują wskaźniki na swojego lewego i prawego syna oraz na swojego ojca.

Czas wykonania podstawowych operacji na drzewie binarnym jest proporcjonalny do wysokości drzewa.

2.3 Tablica haszująca

Tablica haszująca zwana również jako mieszająca. Opiera się ona na zwykłej tablicy indeksowanej liczbami - dostęp do danych jest bardzo szybki nie zależy od rozmiaru tablicy ani od położenia elementu. W tablicy stosuje się funkcję mieszającą, która dla danego klucza wyznacza indeks tablicy. W najprostszym przypadku wartość funkcji mieszającej, obliczona dla zadanego klucza wyznacza dokładnie indeks szukanej tablicy. Jeżeli miejsce wskazywane przez obliczony indeks jest puste, to poszukiwanej informacji nie ma w tablicy. W ten sposób wyszukiwanie elementu ma złożoność czasową $O(1)$. Jednak w sytuacji tej pojawia się problem kolizji, to znaczy przypisania przez funkcję mieszającą tej samej wartości dwóm różnym kluczom.

3 Wykres i tabela porównania tablic asocjacyjnych



Rysunek 1: Wykres pomiaru czasu od zadanej liczby próbek

Tablica 1: Porównanie czasów dostępu do elementu w tablicy asocjacyjnej.

Liczba elementów	Tab przy użyciu listy	Tab przy użyciu drzewa	Tab mieszająca
10	26,9	22,7	23,7
100	33,7	23,4	26,2
1000	38,0	25,0	25,4
10000	59,3	28,8	29,0
100000	94,2	29,9	49,0
1000000	135,8	34,1	161,2

4 Wnioski

Po przeprowadzeniu symulacji zauważyć możemy, że najgorszy czas dostępu uzyskamy przy wykorzystaniu listy z STL. Początkowo tablica haszująca ma podobny czas jak tablica zbudowana na drzewie binarnym. Wraz ze wzrostem rozmiaru problemu najszybszy dostęp możemy uzyskać przy wykorzystaniu binarnego drzewa poszukiwań, jego złożoność obliczeniowa wynosi $\log n$.

Tablica z wykorzystaniem listy również ma złożoność obliczeniową $\log n$ jednak jest ona nieco wolniejsza niż BST. Czas dostępu do elementu nie wzrasta liniowo tak jakby mogło się wydawać gdyż wyszukujemy binarnie.

Jeśli w tablicy haszującej elementów jest co najwyżej tyle ile wynosi rozmiar tablicy, wtedy złożoność obliczeniowa dostępu do elementu jest stała, $O(1)$. Gdy elementów jest więcej zaczynają występować kolizje. Powoduje to, że pod jednym indeksem znajduje się kilka elementów i w celu znalezienia klucza używam wyszukiwania liniowego. W efekcie czas dostępu przestaje być stały.

Tablica haszująca jest najkorzystniejszym rozwiązaniem tablicy asocjacyjnej tylko gdy liczba elementów nie przekracza rozmiaru tablicy. W przeciwnym wypadku złożoność obliczeniowa jest gorsza od pozostałych implementacji.