Sprawozdanie z ćwiczenia laboratoryjnego V
Tablica Asocjacyjna

Bartłomiej Ankowski

17.04.2015

Spis treści

1 Wstęp

Celem zadania było zamodelowanie i zaimplementowanie tablicy asocjacyjnej, opartej na bazie tablicy mieszającej, oraz przedyskutowanie złożoności obliczeniowej operacji odczytu danych dla podanego klucza.

2 Realizacja

Tablica Asocjacyjna jest strukturą danych przechowująca Pary(Unikatowy klucz oraz Wartość) i umożliwa dostęp do wartości poprzez podanie Klucza. Została ona zrealzowana na bazie tablicy mieszające, która wykorzystuję funkcję haszującą do wyliczenia ideksu dla unikatowego klucza.Tablca

mieszająca została zrealizowana przy pomocy struktury danych przygotowanej na poprzednich laboratoriach.Użyty został Stos tablicowy, pracujący na stworzenej strukturze Element, na którą składa się łańcuch znakowy oraz liczba całkowita.Tablica mieszająca zawiera pole, na które składa się tablica wskaźników na Stos tablicowy. Kluczowa dla tej realizacji jest funkcja haszująca, której algorytm jest oparty na dodawaniu do siebie kolejnych znaków klucza.W celu zmniejszenia prawdopodobieństwa wystąpienia kolizji, przed każdym dodawaniem aktualna wartość jest poddawana przesunięciu bitowemu «. W celu uniknięcia otrzymania indeksu większego, niż rozmiar statycznej tablicy mieszającej,otrzymaną sumę poddajemy operacji dzielenia z resztą przez rozmiar tablicy. Ważne przy tym jest, aby rozmiar tablicy był liczbą pierwszą, pozwala to uniknąć zbyt częstego otrzymywania zerowego indeksu jako wynik haszowania.

3 Występowanie kolizji

Idealna funkcja haszująca nie jest możliwa to zrealizowania,nie możliwe jest uzyskanie różnowartościowej funkcji w każdym przypadku.Fakt ten prowadzi do powstawanie kolizji, czyli sytulacji w której dla dwóch kluczy przypisywany ten sam indeks. W takim przypadku może dojść do próby zapisania danych w miejscu, gdzie już zostały zapisane informację. W przypadku realizacji tablicy haszującej istnieje kilka sposobów na rozwiązanie tego problemu. W pierwszej kolejności nalezy się jednak skupić na jak najlepszym skonfigurowaniu funkcji haszującej, tak aby przedział losowanych liczby był jak najszerszy. Drugim sposobem jest umieszczenie danych w innym miejscu niż wskazywałby na to wynik haszowania. Nalezy przy tym zachować określony porządek, tak aby była możliwość prawidłowego odczytania zapisanych danych.

4 Złożoność obliczeniowa

W tym przypadku realizacji wybrano metodę łańcuchową(linkowania). W przypadku wystąpienia kolizji, wartość jest dodawana na koniec stosu, utworzonego dynamicznie na odpowiednim indeksie tablicy mieszającej. W pesymistycznym przypadku złożoność wyosi $O(n)$ i występuje w przypadku, gdy funkcja haszująca zwróci ten sam indeks dla różnych kluczy, czyli wystapi

kolizja.Może to doprowadzić do sytuacji, gdy szukany przez nas element trafi na sam koniec stosu ,znajdującego sie pod danym indeksem tablicy mieszającej. Zatem chcąc odczytać element będacy na końcu danego stosu występuje konieczność sprawdzania zgodonosci klucza każdego z elementów z kluczem poszukiwanym, co znacznie wydłuża czas odczytu.

4.1 Przypadek $O(n)$

Zakładając, że średni czas odczytu wyraża się wzorem $T(n) = a \cdot n$,
gdzie a - czas pojedyńczego odczytu.
W przypadku, gdy szukany przez nas element znajduję się na $n-tym$ miejscu danego stosu, czyli wystąpiło n kolizji.
Trzeba zatem wykonac n porównań kluczy, aby znaleźć wartość pasująca do klucza poszukiwanego.
Implikując złożoność obliczeniowa przypadku pesymistycznego wynosi $O(n)$.

4.2 Przydanek $O(1)$

Zakładając, że średni czas odczytu wyraża się wzorem $T(n) = a \cdot n$,
gdzie a - czas pojedyńczego odczytu.
W przypadku, gdy szukany przez na element jest na pierwszym miejscu stosu,czyli nie wystąpiła do tej pory żadna kolizja, związana z danym indeksem.
Trzeba zatem wykonać $n = 1$ porównań kluczy, aby znaleźć wartość pasująca do klucza poszukiwanego.
Wzór sprowada się do postaci $T(n) = a$, co składa się na złożoność $O(1)$.

Analogicznie wygląda sytuacja przy próbie zapisu wartości.

5 Wyniki pomiarów

Testy zostały przeprowadzone dla dwóch przypadków. W pierwszym rozmiar tablicy mieszającej wynosił 1000033, w drugim wynosił 1.
Testy polegały na pomiarze czasu całkowitego odczytu pewnej ilości danych, czas ten był dzielony przez ilość danych, co ukazywało średni czas pojedynczego odczytu dla określonej porcji danych.

Dla każdej ilości danych test był wykonywalny 10 krotnie.

5.1 Przypadek $O(1)$

Wyniki otrzymane dla Rozmiaru tablicy mieszającej równego 1000033:

Ilość danych[n]	Czas[ms]
100	0,000102
1000	0,000092
10000	0,000129
100000	0,000188
1000000	0,000321

Tablica 1: Przypadek O(1)

Rysunek 1: Przypadek O(1)

5.2 Przypadek $O(n)$

Wyniki otrzymane dla Rozmiaru tablicy mieszającej równego 1

Ilość danych[n]	Czas[ms]
1	0,00050
10	0,00036
50	0,00118
100	0,00271
500	0,01214
1000	0,02796
5000	0,12061
10000	0,28304

Tablica 2: Przypadek O(n)

Rysunek 2: Przypadek O(n)

6 Wnioski

- Tablica Asocjacyjna zapewnia bardzo szybki dostęp do potrzebnych danych przy pomocy użycia Klucza. Może być wykorzystywana przy modelowaniu programów pracujących jako słownik, książka telefoniczna czy każdej innej bazy danych.

- Wyniki, jakie otrzmano podczas testowania struktury, są zgodne z oczekiwaniami.
  Minimalny wzrost czasu pojedynczego odczytu w przypadku złożoności $O(1)$ wiąże się z pewnością z faktem wzrostu prawdopodobieństwa wystąpienia kolizji przy dodawaniu coraz większej ilości danych. Testowanie przypadku, gdy rozmiar tablicy haszującej wynosi 1, dało wynik zgodny z teoretycznym.

- Kluczowe dla wzrostu wydajności programu (skrócenia czasu pojedyń-

czego odczytu) ma prawidłowe skonfigurowanie funkcji haszującej.
W przypadku testów nad tą implementacją, drobne zmiany w tej funkcji zapewniły 2-krotny spadek czasu średniego odczytu w przypadku testów, dla ilości danych równych 1 mln.