

Sprawozdanie z ćwiczenia laboratoryjnego
„Implementacja tablicy asocjacyjnej”.

Karolina Morawska

20.04.2014

1 Wstęp

Celem ćwiczenia było przetestowanie struktury danych jaką jest tablica asocjacyjna , która pozwala dostać się do danych przy użyciu klucza. Mamy różne możliwości zaimplementowania. Ja wybrałam:

- Tablice haszującą
- Drzewo przeszukiwań binarnych

1.1 Tablica haszująca

Zaletą takich struktur jest bardzo szybki dostęp do danych i nie zależy on od rozmiaru tablicy ani położenia elementu. Podstawową wadą tablic mieszających jest duża złożoność pesymistyczna wyszukiwania, wynosząca $\mathcal{O}(n)$ co widać na 1 wykresie. Ponadto kosztowne może być także obliczanie wartości dobrej funkcji mieszającej. Kolejna wada wiąże się z architekturą współczesnych procesorów, które wykorzystują pamięć podręczną. Ponieważ pamięć podręczna przyspiesza odwołania do komórek pamięci operacyjnej, gdy są one zgrupowane blisko siebie, zastosowanie tablicy mieszającej dla zbyt małej liczby elementów może być wolniejsze niż zastosowanie zwykłej tablicy przeszukiwanej sekwencyjnie.

1.2 Drzewo przeszukiwań binarnych

Podstawowe operacje na drzewach poszukiwań binarnych wymagają czasu proporcjonalnego do wysokości drzewa. W pełnym (tzw. zrównoważonym) drzewie binarnym o n węzłach takie operacje działają w najgorszym przypadku w czasie $\mathcal{O}(\lg(n))$, przedstawia to 4 wykres .Jesli jednak drzewo składa się z jednej gałęzi o długości n , to te same operacje wymagają w pesymistycznym przypadku czasu $\mathcal{O}(n)$.

2 Wyniki przeprowadzonych testów

2.1 Tablica haszująca

N	Czas [ms]
100	0
1000	0
10000	0,035
100000	0,5
1000000	4,142

Rysunek 1: Tabela zawierająca pomiar czasu wyszukiwania elementu dla tablicy haszującej.



Rysunek 2: Pomiar czasu wyszukiwania elementu dla tablicy haszującej.

2.2 Drzewo przeszukiwań binarnych

N	Czas[ms]
100	0
1000	0
10000	0,33
100000	0,444
1000000	4,89

Rysunek 3: Tabela zawierająca pomiar czasu wyszukiwania elementu dla drzewka przeszukiwań binarnych.



Rysunek 4: Pomiar czasu wyszukiwania elementu dla drzewka przeszukiwań binarnych.



Rysunek 5: Pomiar czasu wyszukiwania elementu dla dwóch algorytmów. W kolorze niebieskim zaznaczono pomiar dla drzewa binarnego ; kolor czerwony tablica haszująca.

3 Wnioski

- Przedstawione przeze mnie algorytmy cechują się szybkim czasem działania dlatego służą one między innymi do tworzenia słowników.
- Dla danych rzędu miliona czas wyszukiwania w każdej z powyższych struktur jest praktycznie niezauważalny dla pojedynczego wyszukiwania.
- Z wykresu obu algorytmów widać że czas wyszukiwania elementu w tablicy haszującej jest szybszy niż w drzewie przeszukiwań binarnych.
- Kształt wykresu tablicy haszującej powinien być stały. W moim przypadku widać że losowo dobrane elementy już przy 100 000 powodują bardzo szybki wzrost, prawdopodobnie jest to wina funkcji haszującej i powstawaniu kolizji aby tego uniknąć powinno się wykorzystanie metody haszowania kukułczego, rozwiązuje problem kolizji poprzez zastosowanie dwóch tablic i dwóch odpowiadających im funkcji haszujących. Dopóki nie występuje kolizja, dodawane elementy są umieszczane w pierwszej tablicy pod indeksem wyznaczonym przez pierwszą funkcję mieszającą. Jeśli jednak wystąpi kolizja (w miejscu wyznaczonym przez pierwszą funkcję już znajduje się inny obiekt), to wstawiany element jest umieszczany w drugiej tablicy na pozycji wyznaczonej przez drugą funkcję. Jeśli pod tamtym indeksem także znajdował się jakiś obiekt, to zostaje on stamtąd usunięty i dla niego rekurencyjnie zostaje uruchomiona procedura wstawiania, przy czym tym razem zostanie on na siłę wstawiony do pierwszej tablicy. Proces ten jest powtarzany do momentu, w którym przy wstawianiu elementu nie wystąpi kolizja. W przypadku zapętlenia się algorytmu, losowane są nowe funkcje haszujące i wszystkie elementy tablicy

zostają ponownie przemieszane. Jeśli został osiągnięty ustalony współczynnik wypełnienia, to przed wybraniem nowych funkcji należy powiększyć rozmiar tablicy mieszającej.

Haszowanie kukulcze gwarantuje odczyt elementu z tablicy w czasie stałym (gdyż wymagane jest jedynie sprawdzenie dwóch indeksów), a przy losowaniu funkcji mieszających z odpowiedniej rodziny, oczekiwany zamortyzowany koszt wstawienia elementu jest również stały.

Bibliografia

- [1] *[http : //pl.wikipedia.org/wiki/Binarne_drzewo_poszukiwan](http://pl.wikipedia.org/wiki/Binarne_drzewo_poszukiwan),*
- [2] *[http : //pl.wikipedia.org/wiki/Tablica_mieszaj](http://pl.wikipedia.org/wiki/Tablica_mieszaj)*