

Sprawozdanie

Daniel Majchrzycki, 218476

1. Opis programu:

Program implementuje drzewo binarne w postaci drzewa czerwono-czarnego. Wykonuje na nim operację dodawania:

- a) 10
- b) 100
- c) 1 000
- d) 10 000
- e) 100 000
- f) 1 000 000
- g) 10 000 000

losowych liczb. Oraz poszukiwania losowej liczby.

2. Wyniki

a) Dodawanie

Rozmiar problemu	Czas[ms] ładowania wszystkich elementów	Średni czas jednej operacji[ms]
10	< 1	< 0.001
100	< 1	< 0.001
1 000	1	0.001
10 000	14	0.0014
100 000	76	0.00076
1 000 000	1143	0.001143
10 000 000	18829	0.0018829

Tabela 1) Wyniki pomiarów dodawania elementów do drzewa czerwono-czarnego



Wykres 1) Zależność czasu obliczeń od rozmiaru problemu

Z otrzymanych danych trudno stwierdzić jaką złożoność ma algorytm. Prawdopodobnie jest nią $O(\log n)$ wraz z n dążącym do 10^7 . W przypadku mniejszych liczb algorytm nie działa prawidłowo (nie zachowuje przewidywanej złożoności

obliczeniowej). Problemem może być pomiar czasu o małej dokładności powodujący przeszacowanie czasu działania programu.

b) wyszukiwanie

w przypadku wyszukiwania liczb liczba operacji do odnalezienia liczby określona jest wzorem $O(h)$

gdzie $h = 2\log(n + 1)$. Nawet dla ilości liczb 10^7 wymaga to niewielu operacji porównań. Program poszukujący liczby w drzewie zwracał wartość za każdym razem w czasie mniejszym niż dokładność zegara(*ms*).

Uniemożliwiło to zbadanie czy algorytm zachowuje złożoność $O(\log n)$. Korzystając z debuggera można prześledzić krok po kroku jak zachowuje się algorytm. Pozwala to potwierdzić poprawność działania programu.

3. Wnioski:

Zaimplementowany algorytm działał poprawnie dla liczb $10 - 10^7$. Posługując się debuggerem można przejrzeć zawartość kolejnych węzłów drzewa aby upewnić się o ich odpowiednim położeniu.

Użyta implementacja oparta o wskaźniki powoduje alokację dużej ilości pamięci koniecznej dla utrzymania drzewa. Wynika to z konieczności przechowywania wskaźników na 3 inne węzły oraz koloru aktualnego węzła. Nie jest to pamięciowo efektywna metoda przechowywania danych.

Główny atut drzewa binarnego, czyli bardzo krótki czas wyszukiwania danych został zachowany.