

Dodawanie i szukanie w drzewie binarnym

218582

2 maja 2016

1 Wstęp

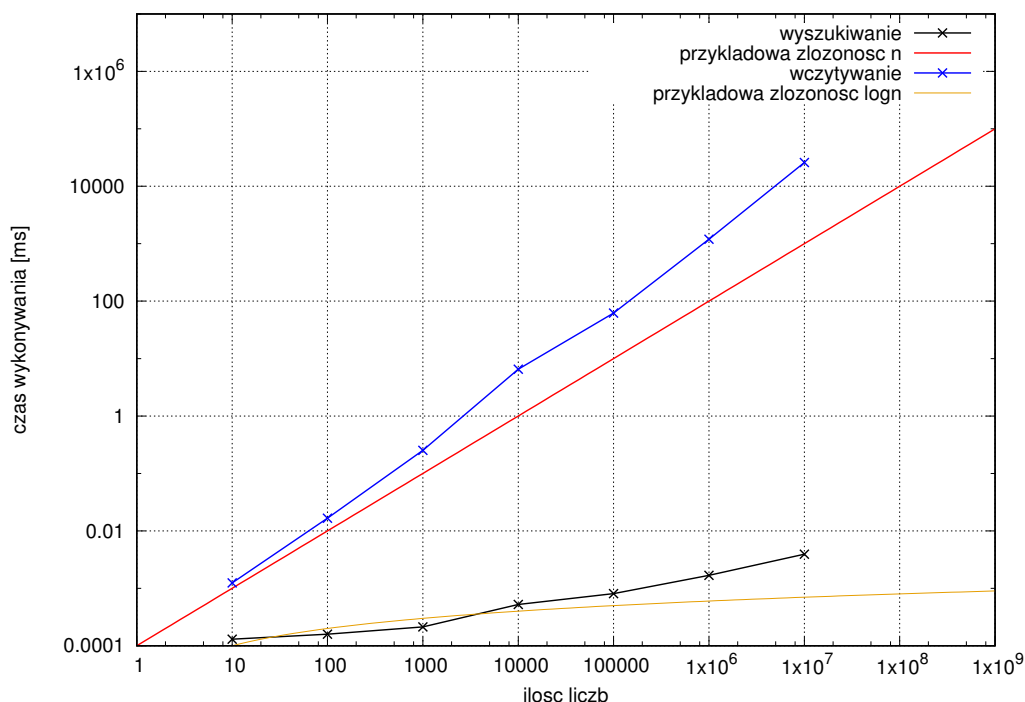
Celem zadania było zmierzenie czasu wyszukiwania klucza w drzewie czerwono-czarnym. Obecnie jednak nie udało się zaimplementować pełnego drzewa czerwono-czarnego. Metoda dodawania klasy tworzy drzewo binarne niezbalansowane. W skrajnym przypadku takie drzewo może zredukować się do listy i wyszukiwanie w nim może mieć złożoność $O(n)$. Oczekiwana złożoność wyszukiwania w obecnie zaimplementowanym drzewie binarnym to $O(\log(n))$, a dodawania - $O(n)$.

Po zaimplementowaniu drzewa czerwono-czarnego do poniższego sprawozdania zostaną dodane nowe wyniki pomiarów oraz porównanie złożoności obu implementacji.

Losowość elementów wpisywanych do drzewa zapewnia użycie funkcji `clock()` z biblioteki `ctime` w `srand()`, która zwraca inną wartość dla każdego taktu procesora.

2 Wyniki

Zgodnie ze standardową procedurą, przeprowadzono testy dla kolejnych rozmiarów problemów. Wyniki prezentuje wykres 1.



Rysunek 1: Działanie wyszukiwania i wpisywania elementów do drzewa niezbalansowanego

3 Wnioski

Zgodnie z wykresem na Rys. 1, można zauważyć, że dla mniejszych rozmiarów złożoność operacji wyszukiwania i dodawania elementów do drzewa pokrywa się z teoretyczną. Jednakże im większa ilość elementów tym złożoność empiryczna zaczyna bardziej odbiegać od teoretycznej. Prawdopodobnie wynika to z wyżej wymienionego faktu - braku pełnej implementacji drzewa czerwono-czarnego.