B Drzewo

Spis treści

[OPIS ALGORYTMU 4](#_Toc406977090)

[OPIS ZADANIA 5](#_Toc406977091)

[PLIK WEJSCIOWY 6](#_Toc406977092)

[PLIK WYJŚCIOWY 7](#_Toc406977093)

[UWAGI 9](#_Toc406977094)

Paweł Borawski

# Opis struktury

B-drzewo – [drzewiasta](http://pl.wikipedia.org/wiki/Drzewo_(informatyka)) [struktura danych](http://pl.wikipedia.org/wiki/Struktura_danych), przechowująca klucze w pewnym porządku i powiązane z nimi dane, używana przede wszystkim w systemach [baz danych](http://pl.wikipedia.org/wiki/Baza_danych). Popularniejsze w zastosowaniach [bazodanowych](http://pl.wikipedia.org/wiki/Baza_danych) i [systemach plików](http://pl.wikipedia.org/wiki/System_plik%C3%B3w) są B+drzewa, które są szczególnym przypadkiem B-drzew, przechowującym dane tylko w liściach.

Głównym pomysłem zastosowanym w B-drzewach jest struktura wewnętrznego węzła. Każdy węzeł może posiadać od M do 2 \* M + 1 węzłów potomnych, gdzie M to rząd B-drzewa; wyjątkiem jest korzeń, który może posiadać od 0 do 2 \* M + 1 węzłów potomnych. Te założenia gwarantują, że wysokość drzewa zawierającego n kluczy będzie niska, rzędu \log_M{n}, co też powoduje, że [asymptotyczna](http://pl.wikipedia.org/wiki/Asymptotyczne_tempo_wzrostu) [złożoność czasowa](http://pl.wikipedia.org/wiki/Z%C5%82o%C5%BCono%C5%9B%C4%87_obliczeniowa) operacji podstawowych: wyszukiwania, wstawiania i kasowania kluczy jest rzędu O(\log_M{n}).

Niska wysokość drzewa powoduje, że liczba węzłów, które trzeba odczytać bądź zapisać, jest niewielka. W praktycznych zastosowaniach, w których informacje przechowywane są na [dyskach twardych](http://pl.wikipedia.org/wiki/Dysk_twardy) bądź płytach [CD](http://pl.wikipedia.org/wiki/P%C5%82yta_kompaktowa)/[DVD](http://pl.wikipedia.org/wiki/DVD) ma to fundamentalne znaczenie, bowiem czasy dostępu do tych urządzeń są dużo większe niż do pamięci wewnętrznej komputera i dominują w całkowitym czasie wykonywania operacji na danych (czasy dostępu do pamięci komputera rzędu mikro- lub setek nanosekund, natomiast do współczesnych dysków twardych to kilka milisekund - czyli 3-4 rzędy wielkości więcej). Z kolei zlokalizowanie odpowiedniego klucza bądź potomka w węźle wczytanym do pamięci wewnętrznej jest dużo szybsze, nawet jeśli rząd drzewa jest duży.

# OPIS ZADANIA

Moim zadaniem było zaimplementowanie b drzewa oraz implementacja podstawowych operacji na tym drzewie (insert, delete, search).

# PLIK WEJSCIOWY

Format pliku wejściowego:

3  
a b c d e f  
insert  
z

# PLIK WYJŚCIOWY

Wynik:

Drzewo przed operacją insert:

c -->

c -->a b

c -->d e f

Drzewo po wykonaniu operacji insert:

c -->

c -->a b

c -->d e f z

# UWAGI

Czasami nie udaje się operacja usunięcia korzenia. Ciężko jest mi namierzyć przyczynę problemu.