

# Laboratorium AiSD

## Operacje na B-drzewach

B-drzewa są parametryzowane wartością  $t$  – minimalnym stopniem węzła. Każdy węzeł, z wyjątkiem korzenia, zawiera minimalnie  $t - 1$  kluczy. Maksymalna liczba kluczy wynosi  $2t - 1$ . Klucze w węźle uporządkowane są rosnąco, a poddrzewo znajdujące się między kluczami  $K_n$  i  $K_{n+1}$  zawiera klucze  $k$  spełniające nierówność:  $K_n \leq k \leq K_{n+1}$ , co jest rozwinięciem własności drzewa BST.

Operacje wykonywane na B-drzewie gwarantują jego zbalansowanie, gdyż liście zawsze znajdują się na tym samym poziomie.

W książce Cormena można znaleźć prostsze wersje poniższych algorytmów, które wymagają pojedynczego przejścia po drzewie. Działają one zapobiegawczo, nadając drzewu taką strukturę, że będzie ono poprawne po danej operacji. **Tutaj przedstawiono algorytmy w wersji z naprawianiem, gdzie przywracanie własności B-drzewa następuje PO wykonaniu operacji.**

### 1. Wyszukiwanie

Operacja wyszukiwania klucza  $k$  przebiega w następujący sposób:

- 1) Jeżeli korzeń jest pusty, wartość nie występuje w drzewie. Koniec.
- 2) Wpp, znajdź miejsce wśród kluczy, gdzie  $k$  powinno znajdować się w węźle:
  - a. Jeśli występuje, zwróć ją. Koniec.
  - b. Wpp:
    - i. Jeżeli węzeł jest liściem, wartość nie występuje w drzewie. Koniec.
    - ii. Wpp, zejdź do poddrzewa, w którym wartość powinna się znajdować i wróć do 2).

## 2. Wstawianie

Operacja wstawiania wymaga pomocniczej operacji podziału węzła wykonywanej dla węzłów pełnych - o  $2t - 1$  kluczy.

Przy podziale, środkowy klucz  $K$  węzła jest przesuwany do rodzica, zaś druga połowa kluczy i dzieci węzła trafia do nowo utworzonego prawego brata  $W_R$ . Pierwsza połowa węzłów i kluczy pozostaje w dzielonym węźle  $W$  (teraz już jako  $W_L$ ).

W wyniku podziału, węzeł oraz jego powstający brat mają minimalną ( $t - 1$ ) liczbę kluczy.

Operacja wstawiania klucza  $k$  przebiega w następujący sposób:

- 1) Jeśli korzeń jest pusty, tworzony jest nowy korzeń i wstawiany jest do niego klucz  $k$ . Koniec.
- 2) Wpp, tak, jak w algorytmie wyszukiwania, znajdowany jest liść, do którego klucz  $k$  powinien trafić. Jeśli na ścieżce od korzenia do liścia znaleziono wartość  $k$ , to kończymy.
- 3) Jeśli węzeł (liść) nie jest pełny, to wstawiamy do niego  $k$ . Koniec.
- 4) Wpp, dokonujemy podziału węzła  $W$  na  $W_L$  oraz  $W_R$  i wstawiamy wartość  $k$  do odpowiedniego węzła. Wartość środkowa  $k'$  jest wstawiana do rodzica węzła  $W$  jak w 3).
- 5) Jeżeli w wyniku podziałów podzielony został również korzeń, tworzony jest nowy korzeń drzewa i wstawiana jest do niego wartość środkowa pochodząca z oryginalnego korzenia. Koniec.

## 3. Usuwanie

Operacja usuwania wymaga dwóch operacji pomocniczych – rotacji oraz złączenia węzłów.

Złączenie węzłów (minimalnych) polega na odwróceniu operacji podziału – klucz  $K$  rozdzielający węzły  $W_L$  i  $W_R$  w rodzicu zostaje wstawiony do  $W_L$ , klucze i dzieci  $W_R$  zostają przepisane na koniec  $W_L$ .  $W_R$  zostaje usunięty.

Rotacja z lewego/prawego brata  $W_B$  węzła  $W$  polega na zabraniu z  $W_B$  klucza maksymalnego/minimalnego, przeniesieniu go do rodzica w miejsce

klucza  $K$  i opuszczenie  $K$  do węzła  $W$ . Do  $W$  przepinane jest skrajne prawe/lewe dziecko  $W_B$ . W wyniku rotacji, węzeł „kradnie” klucz ze swojego brata oraz skrajne dziecko.

Operacja usuwania klucza  $k$  przebiega w następujący sposób:

- 1) Jeżeli korzeń jest pusty – Koniec.
- 2) Wpp, klucz  $k$  jest wyszukiwany w drzewie:
  - a. Jeżeli go nie znaleziono – Koniec.
  - b. Wpp:
    - i. Jeżeli  $k$  znajduje się w liściu, to jest on z niego usuwany. Jeśli w wyniku tej operacji węzeł ma mniej niż  $t - 1$  kluczy i nie jest on korzeniem, to dokonywana jest naprawa:
      1. Jeżeli lewy brat nie jest minimalny, to wykonywana jest rotacja z lewego brata. Koniec.
      2. Jeżeli prawy brat nie jest minimalny, to wykonywana jest rotacja z prawego brata. Koniec.
      3. Wpp, węzeł jest łączony z lewym lub prawym bratem, co może sprawić, że rodzic będzie miał mniej niż minimalną liczbę kluczy. Jeżeli rodzic spełnia warunku B-drzewa, to Koniec. Wpp, rodzic jest naprawiany tak jak w: 1 -> 2 -> 3.
    - ii. Jeżeli  $k$  znajduje się w węźle wewnętrznym, to:
      1. Usuwany jest poprzednik węzła, o ile nie wymaga to naprawy. W przeciwnym przypadku usuwany jest następnik, o ile nie wymaga to naprawy. Zarówno poprzednik, jak i następnik znajdują się w liściu, więc usunięcie przebiega jak w punkcie i.
      2. Jeżeli usunięcie poprzednika i następnika wymaga naprawy, to usuwany jest którykolwiek z nich.
      3. Usunięta wartość  $k'$  zastępuje  $k$  w węźle. Koniec.
  - 3) Jeżeli w wyniku usuwania korzeń stał się pusty, to jest on zastępowany przez swoje jedyne dziecko.

**Krok ii.1 stanowi rozwiązanie idealne, jednak jego pominięcie, tj. neszukanie lepszej (pod względem naprawy) z wartości (poprzednika/następnika) do usunięcia, gwarantuje otrzymanie poprawnego B-drzewa. Z tego powodu, celem uproszczenia, krok ii.1 można pominąć.**