

B-drzewa

Zadanie AiSD.L.BT.01

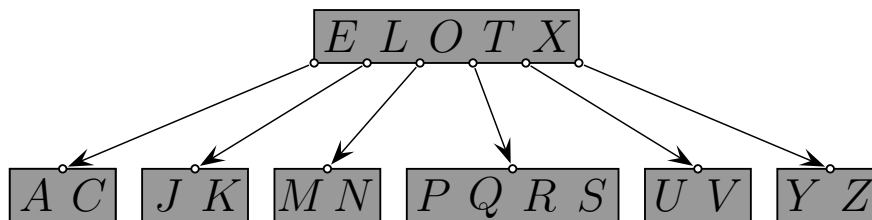
Zilustruj wyniki wstawiania do początkowego pustego B-drzewa stopnia 3 kolejno kluczy $F, S, Q, K, C, L, H, T, V, W, M, R, N, P, A, B, X, Y, D, Z, E, G, J$.

Zadanie AiSD.L.BT.02

Wyjaśnij, jak w B-drzewie wyznaczyć poprzednik danego klucza.

Zadanie AiSD.L.BT.03

Narysuj kolejne B-drzewa powstające w wyniku usuwania z poniższego B-drzewa kolejno kluczy P, C, V, S oraz O .



źródło: Wprowadzenie do algorytm w, Cormen *et al.*, WNT, str. 461, wyd. VII.

Zadanie AiSD.L.BT.04

- (2 pkt.) Zaimplementuj i przetestuj funkcję szukającą elementu w B-drzewie.
- (3 pkt.) Zaimplementuj i przetestuj funkcję wstawiającą element do B-drzewa.
- * (1 pkt) Zaimplementuj i przetestuj funkcję usuwającą element z B-drzewa.

UWAGI.

- Dla uproszczenia przyjmujemy, że klucze są liczbami całkowitymi i ignorujemy ewentualne inne informacje przechowywane z kluczami.
- Węzły drzewa powinny być przechowywane w pliku dyskowym i odczytywane/zapisywane pojedynczo, w miarę potrzeby, jak to zwykle dzieje się w B-drzewach. Rozwiązania, w których węzły drzewa są pamiętane zwyczajnie w pamięci operacyjnej, będą punktowane niżej (o 1 punkt).
- Na stronie przedmiotu w pliku budujB.c (autor dr Paweł Pączkowski) znajduje się program w C ułatwiający (być może) testowanie punktu (a), zawierający funkcję, która buduje przykładowe B-drzewo (węzły w pamięci dyskowej) oraz funkcję, która wyświetla drzewo w następującym stylu.

B-drzewo

```
  D G
 / | \
AB EF HIJ
```

będzie wyświetlone jako

```
  HIJ
  G
  EF
  D
  AB
```

Można też zobaczyć tam (i wykorzystać) przykład zapisu i odczytu węzłów drzewa do pliku.

- Ważnym elementem zadania jest testowanie poprawności zaimplementowanych procedur. Zatem działanie programu, zwłaszcza usuwanie, należy zademonstrować na kilku przykładach odpowiadających różnym przypadkom algorytmu.