

# Algorytmy i Struktury Danych –Drzewa binarne

Laboratorium nr 5 i 6 (4 x 45”) / 9.12.2021

Grupa WCY20IY2S1

Temat:

a) Drzewa BST (Binary Search Tree)

## Zad. nr 1 (4 x 45”)

### Część A. (ocena)

Do pewnego systemu przychodzą liczby całkowite w kolejności losowej z zakresu  $<1,1000>$ . Maksymalna liczba elementów to  $n$ , gdzie  $n$  – podaje użytkownik.

Wykonaj implementację programu, obsługującego drzewo **BST** w oparciu o strukturę listy z dowiązaniami przy założeniach:

- na podstawie kolejno przychodzących elementów buduj sukcesywnie drzewo w kolejności napływających kluczy *key* za pomocą funkcji **insertBST(T, key)**,
- po każdym wprowadzeniu elementu wypisz zawartość drzewa za pomocą funkcji **printBST(T, key, op)**, gdzie:
  - op: 1 – kolejność *inorder* (LVR),
  - op: 2 – kolejność *preorder* (VLR),
  - op: 3 – kolejność *postorder* (LRV),
  - op: 4 – kolejność *level order*,
  - op: 5 – graficzna (tekstowa) reprezentacja drzewa binarnego.
- zbuduj funkcję **countLevBST(T, lev)** – która wyznaczy ilość węzłów na poziomie *lev*,
- zbuduj funkcję **hBST(T)** – która wyznaczy wysokość drzewa (najdłuższą drogę w drzewie od korzenia do liścia w lewym i prawym poddrzewie),
- zbuduj funkcję **removeBST(T, key)**, która pozwala usunąć wskazany przez użytkownika element *key*,
- zbuduj funkcję **checkBST(T)** – która sprawdzi, czy drzewo BST posiada strukturę kopca (true, false),
- zbuduj funkcję **tableBST(T, key)**, która przedstawi reprezentację tablicową drzewa T (o ile to możliwe! (struktura kopca binarnego) – sprawdź wysokość drzewa). Wykorzystaj funkcję **checkBST()**,
- **[dodatkowe]** wykonaj implementację algorytmu optymalizującego wysokość gotowego drzewa BST nie korzystając z własności AVL i algorytmów kolorowania drzewa ( $h = \log_2 n$ ),

### Część B. (ocena dodatkowa)

Podczas zwrotu zadania, każdy student powinien zaprezentować działanie implementowanych, wybranych funkcji „na tablicy” dla dowolnej losowej serii danych.

#### Uwagi dodatkowe:

- Kod programu powinien mieścić się w jednym pliku (włącznie z funkcjami). Przedstawione rozwiązanie powinno być wykonane samodzielnie. W przypadku stwierdzenia kopii wszyscy autorzy identycznych rozwiązań otrzymują ocenę ndst., co wiąże się z koniecznością poprawy ćwiczenia w terminie poprawkowym.
- Do implementacji proszę użyć języka C w standardzie ANSI C lub C++ (bez typów generycznych i bibliotek specjalistycznych)

- Z ćwiczenia laboratoryjnego piszemy sprawozdanie w formie elektronicznej, które jest dodawane do ćwiczenia w oddzielnym pliku w formacie docx, pdf, odt. W sprawozdaniu zamieszczamy odpowiedzi na pytania opisowe i niezbędne obliczenia.
- Gotowe rozwiązania (sprawozdanie i kod programu – dwa pliki niespakowane) przesyłamy za pomocą narzędzia MS Teams do oceny. Ocena po sprawdzeniu będzie dostępna dla każdego studenta w systemie MS Teams.