

Algorytmy i Struktury Danych – Drzewa binarne

Laboratorium nr 7 i 8 (4 x 45”) / 23.12.2021

Grupa WCY20IY2S1

Temat:

- a) Kopce binarne.
- b) Drzewa samoorganizujące się. Drzewo AVL.

Zad. nr 1 (4 x 45”)

Część A. (ocena)

Do pewnego systemu przychodzą liczby całkowite losowe (bez powtórzeń) z zakresu $<1,100>$. Maksymalna liczba elementów to n , gdzie n – podaje użytkownik.

Wykonaj implementację programu, obsługującego drzewo **AVL** w oparciu o strukturę listy z dowiązaniem z dodatkowym parametrem **bf** (*balance factor*) przy założeniach:

- na podstawie kolejno przychodzących elementów buduj sukcesywnie drzewo w kolejności napływających kluczy *key* za pomocą funkcji **insertAVL(T, key)**,
- wykonaj implementację rotacji wierzchołków wg. wskazówek z ćwiczeń/wykładów:
 - **RRrotation**(T, node)
 - **LLrotation**(T, node)
 - **LRrotation**(T, node)
 - **RLrotation**(T, node)
- po każdym wprowadzeniu elementu wypisz zawartość drzewa (key/bf) za pomocą funkcji **printAVL(T, key, op)**, gdzie:
 - op: 1 – kolejność *inorder*(LVR),
 - op: 2 – kolejność *preorder*(VLR),
 - op: 3 – kolejność *postorder*(LRV),
 - op: 4 – kolejność *level order* (poziomami)
- zbuduj funkcję **bfAVL(T, key)** – która wyznaczy współczynnik zbalansowania dla wierzchołka *key*,
- zbuduj funkcję **genBST(T)**, która wygeneruje proste drzewo BST (niezrównoważone), dla drzewa wynikowego zbuduj i sprawdź funkcję **balanceAVL(T)**, która wyważy drzewo w sensie zasady wyważania drzewa AVL.
- zbuduj funkcję **remAVL(T, key)**, która pozwala usunąć wskazany przez użytkownika element *key*,
- zbuduj funkcję **remALL(T)** – usuwającą drzewo z pamięci,

proszę przygotować demo, każdej wykonanej funkcji w programie.

Część B. (ocena dodatkowa)

Podczas zwrotu zadania, każdy student powinien zaprezentować działanie implementowanych, wybranych funkcji „na tablicy” dla dowolnej losowej serii danych. Odpowiedź obejmuje zakres wiedzy: kopce binarne i drzewa AVL.

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 konieczności 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.