

Laboratorium nr 3

Drzewo BST

Celem zadania jest implementacja słownika w postaci „zwykłego” (tzn. ignorując problem wyważania) drzewa poszukiwań binarnych (BST):

- Każdy z węzłów listy posiada składową kluczową, dla której typem danych jest **integer**, oraz referencje do lewego i prawego potomka.
- Należy stworzyć funkcję realizującą:
 - inicjacja pustego drzewa BST;
 - wstawienie do drzewa nowego elementu (wraz z obsługą przypadku, w którym węzeł o zadanej wartości składowej kluczowej już znajduje się w drzewie);
 - wstawienie do listy *X* nowych elementów o wygenerowanych losowo i różnych wartościach składowych kluczowych z zakresu 10 do 1 000 010 (wartość *X* podana jako argument funkcji);
 - wyszukanie w drzewie elementu o podanej jako argument wartości składowej kluczowej (wraz z obsługą przypadku, w którym element nie istnieje);
 - usunięcie z drzewa elementu o podanej jako argument wartości składowej kluczowej (wraz z obsługą przypadku, w którym brak w drzewie węzła o zadanej wartości składowej kluczowej; w przypadku usuwania węzła stopnia 2-go należy zaimplementować wybraną wersję postępowania, odwołujące się do poprzednika lub następnika usuwanego węzła);
 - wyświetlanie drzewa na konsoli (w kolejności od najmniejszej do największej – odwiedziny/przechodzenie drzewa w trybie **inorder**).
- Na liście parametrów formalnych każdej z wymienionych powyżej procedur mogą się pojawić wyłącznie: **referencja (wskaźnik) do korzenia drzewa** oraz **wartość składowej kluczowej** (za wyjątkiem funkcji wstawiania *X* elementów, która zamiast wartości składowej kluczowej zawiera liczbę elementów do wylosowania).

Program po uruchomieniu wczytuje plik wejściowy `inlab03.txt`.

Plik `inlab03.txt` zawiera w pierwszej linii kolejno liczbę elementów do wylosowania **X** i następnie wartości czterech kluczy **k1**, **k2**, **k3**, **k4**.

Następnie wywoływana jest sekwencja funkcji (dalej w funkcji `main()`):

- czas start;
- zainicjuj drzewo;
- usuń element o wartości klucza **k1**; //powinno zwrócić błąd, bo drzewo jest puste
- wstawienie **X** elementów do drzewa;
- wstaw element o wartości klucza **k1**;
- wstaw element o wartości klucza **k2**;
- wstaw element o wartości klucza **k3**;
- wstaw element o wartości klucza **k4**;
- usuń element o wartości klucza **k1**;
- wyszukaj element o wartości **k1**; //powinno zwrócić błąd, bo element został usunięty
- usuń element o wartości klucza **k2**;
- usuń element o wartości klucza **k3**;
- usuń element o wartości klucza **k4**;
- czas stop;

- wypisz czas wykonania.

Przygotowanie e-maila do wysłania:

Uwaga! Kod źródłowy programu (1 plik) po oddaniu prowadzącemu zajęcia laboratoryjne musi zostać przesłany na adres `sdizo@zut.edu.pl`:

- plik z kodem źródłowym musi mieć nazwę: `loginstudenta.sdizo.lab03.main.c` (np. `thyla.sdizo.lab03.main.c`);
- plik musi zostać wysłany z poczty wydziałowej (`wi.zut.edu.pl`);
- nagłówek maila (temat) musi mieć postać: `SDIZO IS1 XXXY LAB03` gdzie XXXY to numer grupy (np. `SDIZO IS1 210C LAB03`);
- w pierwszych trzech liniach pliku z kodem źródłowym w komentarzach musi znaleźć się informacja identyczna z zamieszczoną w nagłówku maila plus imię i nazwisko wysyłającego oraz adres email, z którego wysłał wiadomość;
- email **nie** powinien zawierać żadnej treści (tylko załącznik).