

Instrukcja do laboratorium 3 z ZBP

1 Kontener BST (binary_search_tree)

Stworzyć kontener BST, który udostępnia podstawowy zestaw operacji na drzewie poszukiwań binarnych:

- `bool insert(T x)` — wstawia element do drzewa,
- `bool find(T x)` — sprawdza czy element jest w drzewie,
- `bool erase(T x)` — usuwa element z drzewa,
- `T& min(void)` — zwraca referencję do elementu minimalnego,
- `T& max(void)` — zwraca referencję do elementu maksymalnego.

Należy użyć inteligentnych wskaźników (`shared_ptr`) do implementacji drzewa poszukiwań binarnych. Nie trzeba dostarczać innych metod, ani też oferować możliwości podpięcia własnego alokatora (wystarczy używanie wew. operatora `new`). Nie trzeba dbać o zrównoważenie drzewa (wobec czego do testów lepiej brać dane pomieszane). Pseudokody algorytmów operujących na drzewie są dostępne pod adresem: <http://sun.aei.polsl.pl/~sdeor/index.php?page=aisd>

2 Definiowalny porządek w kontenerze BST

Dodać możliwość przekazania przez użytkownika obiektu funkcyjnego (podobnie jak np. w kontenerze `set`) do ustalenia porządku w drzewie.

3 Iteratory w kontenerze BST*

Dodać obsługę iteratorów (można tylko w jedną stronę) do drzewa BST. Dodatkowe metody do zaprogramowania to: `begin()` oraz `end()`.