

# Drzewa BST

Zadanie zaliczeniowe z ESC/Java2, 20 maja 2011

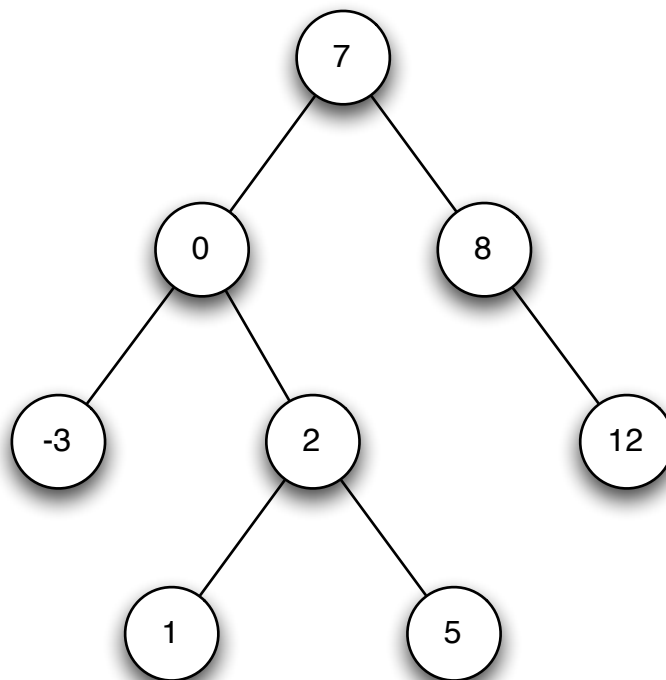
*Termin przesyłania rozwiązań: 3 czerwca 2011*

## 1. Wstęp

Drzewo BST (ang. *binary search tree*) to drzewo binarne (niekoniecznie zrównoważone), gdzie w każdym wierzchołku jest przechowywana liczba całkowita, o tej własności, że dla każdego wierzchołka  $w$ :

- wszystkie liczby przechowywane w lewym poddrzewie  $w$  są mniejsze od liczby przechowywanej w  $w$ , oraz
- wszystkie liczby przechowywane w prawym poddrzewie  $w$  są większe od liczby przechowywanej w  $w$ .

Oto przykład drzewa BST:



Klasa implementująca drzewa BST powinna udostępniać co najmniej następujące metody:

- utworzenie pustego drzewa,
- sprawdzenie, czy podane drzewo jest puste,
- sprawdzenie, czy podana liczba występuje w drzewie,

- dodanie podanej liczby do drzewa (jeśli liczba już występuje w drzewie, to ta metoda nie powinna zmieniać drzewa),
- znalezienie najmniejszej liczby w drzewie,
- usunięcie najmniejszej liczby z drzewa.

Tak zaimplementowanych drzew BST można użyć do posortowania tablicy liczb naturalnych z usuwaniem duplikatów w następujący sposób:

1. Utwórz puste drzewo  $D$ ,
2. Wstaw po kolei wszystkie liczby z tablicy do  $D$ ,
3. Usuwać po kolei najmniejszą liczbę z  $D$  aż  $D$  stanie się puste.

## 2. Zadanie

Napisz w języku Java klasę implementującą drzewa BST jak opisane powyżej. Dodaj do tej klasy metodę statyczną, która używa drzew BST do sortowania ciągów liczb całkowitych.

Następnie uzupełnij swój kod o anotacje JML i użyj ESC/Java2 do zweryfikowania, że procedura sortująca daje w wyniku posortowany ciąg liczb. (Nie musisz weryfikować, że w tym ciągu występują te same liczby co w ciągu wejściowym.)

W tym celu możesz się posłużyć dowolnymi metodami weryfikacji udostępnionymi przez ESC/Java2, w tym niezmiennikami, zmiennymi typu `ghost` itd. Pamiętaj jednak, aby unikać nieuzasadnionego używania anotacji `assume` i `nowarn`.

## 3. Wersja ratunkowa (za 2 pkt.)

Jeżeli zadanie okaże się za trudne, to pomiń weryfikację własności opisanej powyżej, a zadbaj jedynie o to, by program ESC/Java2 przeanalizował kod klasy bez generowania żadnych ostrzeżeń. W tym celu należy m.in. pomóc w sprawdzeniu, że procedura sortująca nigdy nie spróbuje usunąć elementu z pustego drzewa.