

Lista 6

Zadanie 1 (0,4 punktu). Wyznacz liczbę drzew binarnych o czterech węzłach.

Zadanie 2 (0,5 punktu). W pewnym drzewie o n wierzchołkach, trawersowanie w porządku `preorder` odwiedza wierzchołki w tej samej kolejności, co w porządku `inorder`. Co to za drzewo?

Zadanie 3 (0,75 punktu). Rozważmy takie drzewa wyszukiwań binarnych, które mając n węzłów mają najmniejszą możliwą wysokość. Wyznacz pesymistyczną i średnią złożoność algorytmu wyszukiwania elementu w takim drzewie (w zależności od n).

Zadanie 4 (0,75 punktu). Narysuj drzewo T wyrażenia arytmetycznego, które spełnia jednocześnie następujące warunki:

- T ma siedem węzłów, w tym cztery liście,
- Liście przechowują wartości: 2, 3, 5, i 8 (każda wartość pojawia się w jednym z liści, niekoniecznie w tej kolejności),
- Węzły wewnętrzne¹ przechowują operatory ze zbioru $\{+, -, \cdot, /\}$ (każdy operator może wystąpić w drzewie dowolnie wiele razy),
- Wartość wyrażenia opisanego przez T to 16,
- Wartości reprezentowane przez poddrzewa T mogą być ułamkami.

Zadanie 5* (0,1 punktu) Ile jest drzew z Zadania 4 i dlaczego?

Zadanie 6 (0,5 punktu). Uzupełnij klasę `BinaryTree` o metodę `copy(self)`, zwracającą *płytką* kopię drzewa. Płytką kopia drzewa T to kopia korzenia T (nowa instancja `BinaryTree` przechowująca ten sam obiekt, co oryginalny korzeń), którego lewym i prawym poddrzewem jest płytka kopia odpowiednio lewego i prawego poddrzewa korzenia T (lub `None` w wypadku braku danego poddrzewa).

Zadanie 7 (1 punkt) Uzupełnij klasę `BinaryTree` z wykładu:

- Dodaj metodę `height(self)`, zwracającą wysokość drzewa.
- Dodaj metodę `count_leaves(self)`, zwracającą liczbę liści w drzewie.
- Zmodyfikuj metody `preorder(self)`, `inorder(self)` i `postorder(self)` aby zwracały iteratory, zwracające węzły (uwaga: same węzły, nie wartości w węzłach) drzewa zgodnie z odpowiednim porządkiem.

¹Wezeł wewnętrzny to węzeł, który nie jest liściem.