

**Zadanie 1 (14 pkt)**

Liściem w drzewie z korzeniem nazywamy każdy węzeł bez następników (synów).

Niech  $T$  będzie drzewem wyszukiwań binarnych powstałym przez wstawienie, do początkowo pustego drzewa, kolejno  $n$  parami różnych kluczy  $k_1, k_2, \dots, k_n$ . W dalszym ciągu klucze utożsamiamy z zawierającymi je węzłami.

1. (2 pkt)

Zaproponuj liniowy algorytm, który dla danych:  $n$ , ciągu kluczy  $k_1, k_2, \dots, k_n$  i indeksu  $j$ ,  $1 \leq j \leq n$ , sprawdzi, czy węzeł  $k_j$  jest liściem w  $T$ .

2. (3 pkt)

Założmy, że ciąg  $k_1, k_2, \dots, k_n$  jest permutacją liczb  $1, 2, \dots, n$ . Zaproponuj algorytm, który w czasie  $O(n)$  wyznaczy wszystkie liście w drzewie  $T$ .

3. (2 pkt)

Udowodnij, że jeżeli z AVL-drzewa usuniemy wszystkie liście, to otrzymane drzewo będzie nadal AVL-drzewem.

4. (2 pkt)

Ile wynosi minimalna liczba liści w AVL-drzewie o wysokości 2012?

5. (5 pkt)

Założmy, że ciąg  $k_1, k_2, \dots, k_n$  jest permutacją liczb  $1, 2, \dots, n$ , a drzewo wyszukiwań binarnych  $T$  będzie AVL-drzewem. Zaproponuj algorytm, który w czasie liniowym zbuduje drzewo  $T$ , tzn. dla każdego węzła (klucza) określi jego lewego i prawego syna.

**Zadanie 2 (6 pkt)**

Kolejka podwójna, to liniowa struktura danych (elementy kolejki są ustawione w ciąg), na której wykonywane są operacje wstawiania i usuwanie elementów z obu jej końców. Zaproponuj implementację kolejki podwójnej na trzech stosach, w taki sposób, żeby każda operacja na niej była wykonywana w zamortyzowanym czasie stałym.

**Zadanie 3 (bonus za 5 pkt)**

Zaproponuj efektywną strukturę danych dla skończonego ciągu liczbowego  $\alpha$  długości  $n$ , umożliwiającą efektywne wykonywanie on-line następujących operacji:

*Rosnący*:: ile wynosi długość najdłuższego podciągu rosnącego złożonego z kolejnych elementów ciągu  $\alpha$

*Zmien( $i, \Delta$ )*:: do  $i$ -tego elementu ciągu dodaj  $\Delta$

**Uzasadnij poprawność swoich rozwiązań. Każde zadanie oddajemy na oddzielnej kartce.**