ASD - klasówka 2

Zadanie 1 (15 pkt)

Liściem w drzewie z korzeniem nazywamy każdy węzeł bez następników (synów).
Niech T będzie drzewem wyszukiwań binarnych powstającym przez wstawienie, do początkowo pustego drzewa, kolejno n parami różnych kluczy $k_1, k_2, \ldots, k_n$. W dalszym ciągu klucze utożsamiamy z zawierającymi je węzłami.

1. (2 pkt)
   Zaproponuj liniowy algorytm, który dla danych: n, ciągu kluczy $k_1, k_2, \ldots, k_n$ i indeksu j, $1 \leq j \leq n$, sprawdzi, czy węzeł k_j jest liściem w T.

2. (5 pkt)
   Załóżmy, że ciąg $k_1, k_2, \ldots, k_n$ jest permutacją liczb $1, 2, \ldots, n$. Zaproponuj algorytm, który w czasie $O(n)$ wyznaczy wszystkie liście w drzewie T.

3. (1 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)
   Załóżmy, że ciąg $k_1, k_2, \ldots, k_n$ jest permutacją liczb $1, 2, \ldots, n$ i wiemy, że po wstawieniu do początkowo pustego drzewa wyszukiwań binarnych kolejno kluczy k_i, $i = 1, 2, \ldots, n$, otrzymane drzewo T będzie AVL-drzewem (nie wykonujemy żadnych operacji równoważenia). Zaproponuj algorytm, który w czasie liniowym zbuduje drzewo T, tzn. dla każdego węzła (klucza) określi jego lewego i prawego syna.

Zadanie 2 (5 pkt)

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

Rosnący:: ile wynosi długość najdłuższego podciągu rosnącego złożonego z kolejnych elementów ciągu α

Zmień(i, Δ):: do i-tego elementu ciągu dodaj Δ (to może być liczba ujemna)

Pokaż, w jaki sposób efektywnie zainicjować zaproponowaną strukturę danych dla początkowego ciągu α.

Zadanie 3 (bonus za 7 pkt)

Dane jest n-wierzchołkowe drzewo z korzeniem T, z wagami na krawędziach (liczby całkowite). Dla każdego wierzchołka v różnego od korzenia dane są ojciec $p[v]$ w drzewie i waga $w[v]$ krawędzi v–$p[v]$. Przyjmujemy też, że wierzchołki są ponumerowane w porządku "preorder" i utożsamiamy je z tymi numerami – v oznacza zarówno wierzchołek, jak i jego numer.
Zaproponuj algorytm, który w czasie $O(n+k)$ udzieli odpowiedzi na k z góry zadanych zapytań postaci:

Max(u, v):: ile wynosi waga najcięższej krawędzi na ścieżce z u do v?

przy czym wiemy, że w każdym zapytaniu v jest przodkiem u w drzewie.

Uzasadnij poprawność swoich rozwiązań. Każde (pod-)zadanie oddajemy na oddzielnej kartce.