Algorytmy i Struktury Danych

1. (1pkt) Czy można tak zmodyfikować drzewa czerwono-czarne, by operacje *insert, delete, search, minimum, maksimum* nadal wykonywały się w czasie $O(\log n)$, a operacje *następnik*(v) i *poprzednik*(v), gdzie v jest adresem węzła, wykonywane były w czasie $O(1)$?

2. (1pkt) Czy następujące stwierdzenie jest prawdziwe: Jeśli operację wstawiania elementu do 2-3-4 drzewa dokonujemy metodą zstępującą, to żaden 3-wierzchołek nie ma wśród swoich dzieci 3-wierzchołka.

   Przypomnienie terminologii: 2-3-4 drzewa, to B-drzewa, których wierzchołki mogą pamiętеć od 1 do 3 kluczy; 3-wierzchołek, to wierzchołek pamiętający 3 klucze.

3. (1pkt) Napisz procedury obsługujące kopiec Minimaksowy.

4. (2pkt) Niech $h(v)$ oznacza odległość wierzchołka v do najbliższego pustego wskaźnika w poddrzewie o korzeniu v. Rozważ możliwość wykorzystania drzew binarnych, w których zachowana jest następujący warunek zrównoważenia:

   $$h(\text{lewy syn } v) \geq h(\text{prawy syn v}) \text{ dla każdego wierzchołka } v$$

   do implementacji kolejki priorytetowej.

5. (2pkt) Ułóż algorytm, który dla danego ciągu liczbowego znajduje najdłuższy jego podciąg rosnący. Jeśli podciągów rosnących o maksymalnej długości jest więcej niż jeden, Twój algorytm powinien wypisać pierwszy (wg porządku leksykograficznego) z nich. Jaka jest złożoność pamięciowa i czasowa Twojego algorytmu?

6. (2pkt) Ułóż algorytm, który dla danego ciagu liczbowego oblicza ile zawiera on podciągów rosnących o maksymalnej długości. Jaka jest złożoność pamięciowa i czasowa Twojego algorytmu? Przyjmij, że operacje arytmetyczne na dowolnie długich liczbach wykonują się w czasie jednostkowym.

   Czy Twój algorytm wykorzystuje drzewa zbalansowane? Jeśli nie, zastanów się nad możliwością ich zastosowania. Jeśli tak, zastanów się nad możliwością ich wyeliminowania :-).

7. (2pkt) Napisz procedurę *Split*(T, k) rozdzielającą drzewo AVL T na dwa drzewa AVL. Jedno zawierające klucze mniejsze od k i drugie zawierające pozostałe klucze. Jaka jes tzłożoność Twojej procedury?

Zadania dla grupy zaawansowanej

1. (2pkt) Pokaż w jaki sposób można zaimplementować kolejkę priorytetową tak, by operacje na niej wykonywane były w czasie $O(\log\log m))$, gdzie m jest mocą uniwersum, z którego pochodzą klucze.

2. (3 pkt) *Okienkiem* o rozmiarze k w ciągu nazywamy dowolny podciąg kolejnych jego k elementów. Ułóż algorytm, który dla danego ciągu liczbowego A i liczby naturalnej k oblicza najdłuższy podciąg rosnący ciągu A mieszczący się w okienku o rozmiarze k.