Lista zadań. Nr 6. 4 maja 2007

Algorytmy i Struktury Danych

IIUWr. II rok informatyki.

1. (1pkt) Czy można tak zmodyfikować drzewa AVL, 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) Napisz procedury obsługujące kopiec Minimaksowy.

3. (2pkt) Niech $h(v)$ oznacza odległość wierzchołka v do najbliższego pustego wskaźnika w poddrzewie o korzeniu v. Rozważ możliwość wykorzystania drzew binarnych, w których zachowany jest następujący warunek zrównoważenia:

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

   do implementacji kolejki priorytetowej.

4. (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 z nich (wg porządku leksykograficznego). Jaka jest złożoność pamięciowa i czasowa Twojego algorytmu?

5. (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 :-).

6. (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 jest złożoność 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. (2pkt) Bolesną dolegliwością związaną z drzewami AVL jest konieczność poświęcenia dwóch bitów w każdym węźle na pamiętanie współczynnika zrównoważenia. Zastanów się, czy aby na pewno mamy do czynienia z "koniecznością".

Problem

3. (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.