ASD – Klasówka I
19 (godz. 14)-20 (godz. 10.30) grudnia 2007 r.

Uwagi ogólne:
-rozwiązanie każdego zadania jest warte 5 punktów
-rozwiązanie każdego zadania oddajemy na odddzielnej kartce (kartkach)
-rozwiązania powinny być czytelne
-zadania rozwiązujemy samodzielnie
-wolno korzystać z książek, notatek, ważniaka; nie można korzystać z innych zasobów w Internecie
-w zadaniach algorytmicznych należy uzasadnić poprawność swoich rozwiązań

POWODZENIA

Zadanie 1

Opracuj strukturę danych, która pozwala wykonywać następujące operacje:

Ini(k):: inicjacja struktury danych i ustalenie długości krotek liczb całkowitych na k

Insert(<a_1, a_2, ..., a_k>):: dodaje do struktury krotkę <a_1, a_2, ..., a_k>

Min:: podaje najmniejszą leksykograficznie krotkę w strukturze

ExtractMin:: usuwa najmniejszą leksykograficznie krotkę ze struktury

W Twoim rozwiązaniu operacje Insert i ExtractMin powinny by wykonywane w czasie O(log n + k).

Zadanie 2

Udowodnij, że jeśli algorytm sortujący tablicę A[1..n] porównuje i zamienia wyłącznie elementy odległe co najwyżej o 2007 (tzn. jeśli porównuje A[i] z A[j], to |i-j| <= 2007), to jego pesymistyczny czas działania jest co najmniej kwadratowy.

Zadanie 3

Zaproponuj implementację następujących operacji na tablicy liczb naturalnych T[0..n+1], początkowo wypełnionej zerami, w taki sposób, żeby ich koszt zamortyzowany był stały.

Inc(i):: T[i] := T[i] + 1 // zawsze 0 < i < n+1

BlockDec(i):: Jeśli T[i]=0 nic nie rób. W przeciwnym przypadku znajdź najbliższy indeks j taki, że T[i] <> T[j] (jeśli są dwa takie indeksy wybieramy mniejszy z nich). Dla każdego k pomiędzy i oraz j wykonaj T[k] := T[k] – 1.

Zadanie 4

Udowodnij, że dla każdego naturalnego n istnieje drzewo czerwono-czarne o co najmniej n wierzchołkach, które nie jest AVL-drzewem.