Egzamin z ASD, 19.01.2002

Każde zadanie należy oddawać na osobnej, podpisanej kartce. W rozwiązaniach zadań 2, 3, 4 uzasadnij poprawność napisanych przez siebie algorytmów i oszacuj ich złożoność czasową i pamięciową. Powodzenia !

Zadanie 1 (10 punktów)

a) Na początkowo pustym kopcu zaimplementowanym za pomocą zupełnego drzewa binarnego wykonano operacje `Wstaw` dla następujących liczb:

$$5, 9, 20, 4, 18, 25, 3$$

a następnie dwukrotnie wykonano operację `UsuńMax`. Narysuj wygląd kopca traktowanego jako drzewo binarne po każdej z operacji wstawiania i usuwania.

b) Narysuj doskonale wyważone drzewo BST zawierające liczby wstawiane do kopca w punkcie (a). Narysuj także drzewo BST zawierające te same elementy, które nie jest doskonale wyważone.

Zadanie 2 (15 punktów)

Średnicą grafu nazywamy największą odległość pomiędzy dwoma wierzchołkami tego grafu, czyli jeśli $G = (V, E)$, to

$$diam(G) = \max_{v,w \in V} dist(v, w)$$

Napisz algorytm, który obliczy średnicę drzewa z korzeniem, zapisanego w pamięci za pomocą reprezentacji „lewy syn - prawy brat". Można założyć istnienie zmiennej `Korzeń` zawierającej wskaźnik do korzenia drzewa.

Zadanie 3 (20 punktów)

Dany jest ciąg n liczb naturalnych oraz liczba k. Napisz algorytm, który znajdzie k najmniejszych elementów tego ciągu. Należy przyjąć, że n może być bardzo duże (np. $n = 10^9$), w szczególności zbyt duże, aby cały ciąg zmieścił się w pamięci komputera (kolejne elementy ciągu pojawiają się na wejściu). k jest natomiast dużo mniejsze od n (np. $k = 1000$).
Rozwiązanie $O(nk)$: 10 punktów. Rozwiązanie $O(n \log k)$: 20 punktów.

Zadanie 4 (25 punktów)

Niech T będzie tablicą liczb całkowitych o wymiarach $n \times m$. Współrzędnymi pola w tej tablicy jest para (w, k) gdzie $w \in \{1 \ldots n\}$ oznacza numer wiersza, a $k \in \{1 \ldots m\}$ numer kolumny na której znajduje się pole. *Przekątną* tablicy T nazywamy dowolny ciąg współrzędnych pól

$$P = (w_1, k_1), (w_2, k_2), \ldots (w_{n+m-1}, k_{n+m-1})$$

taki, że $(w_1, k_1) = (1, 1)$, $(w_{n+m-1}, k_{n+m-1}) = (n, m)$ oraz dla dowolnego $i = 1, 2, \ldots n + m - 2$ zachodzi albo $w_{i+1} = w_i + 1$ i $k_i = k_i$ albo $w_i = w_i$ i $k_{i+1} = k_i + 1$.

Innymi słowy, przekątna tablicy T to trasa od pola $(1, 1)$ do pola (n, m) taka, że w każdym ruchu możemy albo przejść do następnego wiersza (w dół) albo do następnej kolumny (w prawo).

Wagą przekątnej P nazywamy sumę liczb znajdujących się na tej przekątnej, czyli

$$Waga(P) = \sum_{i=1}^{n+m-1} T[w_i, k_i]$$

Napisz algorytm, który:

1. Obliczy największą spośród wag przekątnych w tablicy (15 punktów)
2. Znajdzie przekątną o największej wadze (10 punktów)