ASD gr.13

Kolokwium z algorytmów i struktur danych 19-11-2014

Imię i nazwisko: …………………………………………………. Nr indeksu: ……………

Zadanie 1. (5 pkt.)

Dana jest tablica $T[1 \dots n]$ zawierająca liczby naturalne. Napisz pseudokod algorytmu, który obliczy, ile razy występują w niej wartości mniejsze od x które nie należą do podtablicy$P = T[y \dots z]$ gdzie $1 \le y \le z \le n$, przy czym tablicę przeglądamy od indeksu 1 do indeksu n. Jeżeli w tablicy nie ma liczb mniejszych od x z poza podtablicy P, należy wyznaczyć iloczyn liczb z podtablicy P. Parametrami wejściowymi algorytmu powinny być wartości: (T, n, x, y, z)

Zadanie 2. (6 pkt.)

Dana jest tablica $T[1 \dots n]$ zawierająca pewne liczby. Napisz algorytm rekurencyjny wyznaczający sumę elementów w tej tablicy, korzystający z następującej własności: suma wszystkich elementów w tablicy jest sumą: wartości sumy elementów od 1 do $\left\lfloor\frac{n}{3}\right\rfloor$, wartości sumy elementów od $\left\lfloor\frac{n}{3}\right\rfloor + 1$ do $\left\lfloor\frac{2n}{3}\right\rfloor$, oraz wartością sumy elementów od $\left\lfloor\frac{2n}{3}\right\rfloor + 1$ do n. Podaj równanie rekurencyjne opisujące czas działania tego algorytmu i rozwiąż je (podaj dokładne oszacowanie asymptotyczne).

Zadanie 3. (5 pkt.)

Stosując metodę **programowania dynamicznego**, napisz pseudokod algorytmu znajdującego dla danych liczb naturalnych *n, k* wartość funkcji $f(n,k)$ określonej wzorami:

$$f(n,k) = \begin{cases} 2k & dla\ n = 1 \\ 3n & dla\ k = 1\ i\ n > 1 \\ 2f(n-1,k) + (f(n,k-1) mod\ 3) & dla\ n,k > 1 \end{cases}$$

Jaka jest złożoność tego algorytmu? Odpowiedź uzasadnij.

Zadanie 4. (3 pkt.)

a. Korzystając z definicji, sprawdź, czy prawdziwe jest oszacowanie: $2\,n^2 + 4\,n - 8 = \Theta(n^2)$
b. Określ ograniczenie asymptotyczne funkcji $T(n)$ danej wzorem: $T(n) = 16T\left(\frac{n}{4}\right) + n^4$
c. Określ ograniczenie asymptotyczne funkcji $T(n)$ danej wzorem: $T(n) = T(n-1) + n + 1$

Zadanie 5. (6 pkt.)

Dany jest *n* trójek dodatnich liczb rzeczywistych $(a_1, b_1, c_1), \dots, (a_n, b_n, c_n)$, reprezentujących długości boków *n* prostopadłościanów (do a_1 odwołujemy się w następujący sposób $T[1].a$). Napisz pseudokod algorytmu sortującego te prostopadłościany w porządku nierosnącym względem wartości ich objętości, dowolna metodą **za wyjątkiem** sortowania bąbelkowego. Wynik działania algorytmu powinien być wypisany w postaci posortowanego nierosnąco ciągu liczb oznaczających objętości poszczególnych prostopadłościanów. Np. dla danych wejściowych $[(1,2,3), (4,5,2), (3,4,2)]$ prawidłowym wynikiem jest $[40,24,6]$. Jaka jest złożoność tego algorytmu? Odpowiedź uzasadnij.

Twierdzenie o rekurencji uniwersalnej. Niech $a \ge 1$ i $b > 1$ będą stałymi, niech $f(n)$ będzie pewną funkcją i niech $T(n)$ będzie zdefiniowane dla nieujemnych liczb całkowitych przez rekurencję

$$T(n) = aT\left(\frac{n}{b}\right) + f(n),$$

gdzie $\frac{n}{b}$ interpretujemy jako $\left\lfloor\frac{n}{b}\right\rfloor$ lub $\left\lceil\frac{n}{b}\right\rceil$. Wtedy funkcja $T(n)$ może być ograniczona asymptotycznie w następujący sposób:

1. Jeśli $f(n) = O\left(n^{log_b a - \varepsilon}\right)$ dla pewnej stałej $\epsilon > 0$, to $T(n) = \Theta\left(n^{log_b a}\right)$.
2. Jeśli $f(n) = \Theta\left(n^{log_b a}\right)$ to $T(n) = \Theta\left(n^{log_b a} \lg n\right)$.
3. Jeśli $f(n) = \Omega\left(n^{log_b a + \varepsilon}\right)$ dla pewnej stałej $\epsilon > 0$, oraz $a\,f\left(\frac{n}{b}\right) \le c\,f(n)$ dla pewnej stałej $c < 1$ i wszystkich dostatecznie dużych *n*, to $T(n) = \Theta\left(f(n)\right)$.