Algoritmi şi Structuri de Date (I). Seminar 2: Descrierea în pseudocod a algoritmilor

- Care sunt principalele tipuri de prelucrări? Care prelucrări sunt considerate elementare?
- Cum putem descrie un algoritm?
- Exemple de algoritmi care prelucrează date numerice

Problema 1 Fie n un număr natural nenul. Descrieţi în pseudocod algoritmi pentru:

(a) Determinarea sumei tuturor cifrelor lui n. De exemplu, pentru $n = 26326$ se obţine valoarea 19.

(b) Determinarea valorii obţinute prin inversarea cifrelor numărului n. De exemplu, pentru valoarea 26326 se obţine valoarea 62362.

(c) Determinarea mulţimii tuturor cifrelor ce intervin în număr. De exemplu, pentru valoarea 26326 se obţine mulţimea $\{2, 3, 6\}$.

(d) Determinarea tuturor cifrelor binare ale lui n.

(e) Determinarea tuturor divizorilor proprii ai lui n.

(f) A verifica dacă numărul n este prim sau nu (algoritmul returnează *true* dacă numărul este prim şi *false* în caz contrar).

(g) Determinarea descompunerii în factori primi a lui n. De exemplu pentru $490 = 2^1 \cdot 5^1 \cdot 7^2$ se obţine mulţimea factorilor primi: $\{2, 5, 7\}$ şi puterile corespunzătoare: $\{1, 1, 2\}$.

Problema 2 Fie n un număr natural, x o valoare reală din $(0, 1)$ şi $\epsilon > 0$ o valoare reală pozitivă. Descrieţi în pseudocod un algoritm pentru:

(a) Calculul sumei finite $\sum_{i=1}^{n}(-1)^i x^{2i}/(2i)!$.

(b) Calculul aproximativ al sumei infinite $\sum_{i=1}^{\infty}(-1)^i x^{2i}/(2i)!$ cu precizia ϵ.

Problema 3 Să se afişeze primele N elemente şi să se aproximeze (cu precizia ϵ) limitele şirurilor (cu excepţia şirului de la punctul (d) care nu este neapărat convergent):

(a) $x_n = (1 + 1/n)^n$;

(b) $x_1 = a > 0$, $x_n = (x_{n-1} + a/x_{n-1})/2$;

(c) $x_n = f_{n+1}/f_n$, $f_1 = f_2 = 1$, $f_n = f_{n-1} + f_{n-2}$;

(d) $x_1 = s$, $x_n = (ax_{n-1} + b)\mathrm{MOD}c$, $a, b, c \in N^*$.

Probleme suplimentare

1. Să se descrie în pseudocod şi să se implementeze algoritmul înmulţirii ”à la russe”.

2. Să se descrie în pseudocod şi să se implementeze algoritmul metoda de sortare bazată pe răsturnarea unei subsecvenţe finale a şirului (problema clătitelor).

3. Propuneţi un algoritm care aplicat pentru două şiruri cu acelaşi număr de elemente decide dacă unul dintre şiruri poate fi obţinut din celălalt printr-o singură răsturnare a unei secvenţe finale (în cazul a două stive de clătite înseamnă că una este obţinută din cealaltă prin aplicarea unei singure operaţii de răsturnare)

4. Algoritm care determină numărul de cifre binare egale cu 1 din reprezentarea în baza 2 a valorii naturale nenule n.

5. Algoritm care să determine cifra ce apare cel mai frecvent într-un număr natural nenul n. Dacă sunt mai multe astfel de cifre se vor afişa toate.

6. Algoritm pentru calculul sumei $\sum_{i=0}^{n}(-1)^i x^{2(i+1)}/(2(i+1))!$ şi pentru aproximarea sumei infinite corespunzătoare ($\sum_{i=0}^{\infty}(-1)^i x^{2(i+1)}/(2(i+1))!$).

7. Algoritm pentru a afişa primele N elemente ale şirului lui Fibonacci ($f_1 = f_2 = 1$, $f_n = f_{n-1} + f_{n-2}$ pentru $n \geq 3$). Descrieţi două variante: (i) o variantă de algoritm care utilizează trei variabile de lucru; (ii) o variantă de algoritm care utilizează doar două variabile de lucru.

8. Scrieţi un algoritm care descompune un număr natural n în două numere p şi i astfel: p conţine cifrele pare din n (nu contează ordinea cifrelor) iar i conţine cifrele impare (nu contează ordinea cifrelor). De exemplu pentru $n = 54672$, p poate fi 264 iar i poate fi 75. In cazul în care n nu conţine cifre pare atunci $p = 0$ iar dacă nu conţine cifre impare atunci $i = 0$.

9. Conjectura lui Goldbach afirmă că: ”orice număr par mai mare decât 2 poate fi descompus ca sumă a două numere prime”. Descrieţi un algoritm care ar putea fi utilizat pentru a invalida conjectura (prin descoperirea unui contraexemplu).

10. Propuneţi un algoritm care transfrmă un număr natural prin deplasarea circulară a cifrelor către stânga cu k poziţii. Se presupune că k este mai mic decât numărul de cifre din n şi că nu se pot reţine cifrele numărului într-un tablou (se poate lucra doar cu variabile scalare). De exemplu pentru $n = 461739$ şi $k = 2$ ar trebui să se obţină 173946.