Structuri de date

Tema 1

1. Scrieţi o funcţie recursivă care calculează coeficientul binominial C_n^k în funcţie de C_n^{k-1}. Analizaţi funcţia de operaţii şi funcţia de locaţii şi comparaţi-o cu cazul iterativ. Aceiaşi problemă însă C_n^k va fi scris în funcţie de C_{n-1}^k şi C_{n-1}^{k-1}.

2. Funcţia lui Akerman A(m,n) dată de:

$$A(m,n)=\begin{cases} n+1 \text{ dacă m} = 0 \\ A(m-1,1) \text{ dacă n} = 0 \\ A(m-1, A(m,n-1)), \text{altfel} \end{cases}$$

are proprietatea că pentru valori relativ mici ale lui m şi n are o creştere foarte mare. Scrieţi o funcţie recursivă pentru calculul valorii A(m,n).

3. Fie funcţia $F:\{1,2,\ldots,n\} \rightarrow \{1,2,\ldots,n\}$ bijectivă, fără puncte fixe. Să se scrie un algoritm care determină (dacă există) cea mai mare submulţime $S \subset \{1,2,\ldots,n\}$ cu proprietatea că F(S)=S.

4. Dacă S este o mulţime cu n elemente, atunci mulţimea părţilor lui S este mulţimea tuturor submulţimilor lui S incluzând mulţimea vidă şi mulţimea totală. Scrieţi o procedură recursivă care generează toate submulţimile mulţimii S.

5. „Jocul vieţii„ este o problemă devenită clasică în matematică, introdusă în 1970 de matematicianul englez J.H. Conway. Jocul se desfăşoară pe un careu rectangular nemărginit, în care fiecare celulă poate fi ocupată sau neocupată, celulele ocupate numindu-se vii iar cele neocupate numindu-se moarte; starea celulelor se schimbă de la o generaţie la alta, după următoarele reguli:

- i) vecinii unei celule date sunt cele 8 celule ce o înconjoară pe verticală, orizontală şi diagonală;
- ii) dacă o celulă este vie şi are 0 sau un vecin viu, atunci ea va muri de singurătate în următoarea generaţie;
- iii) dacă o celulă este vie şi are patru sau mai mulţi vecini vii, atunci ea va muri sufocată în următoarea generaţie
- iv) o celulă moartă cu exact trei vecini va deveni vie în generaţia următoare

Sa se scrie un algoritm care generează următoarea configuraţie a careului pornind de la o configuraţie dată, respectând regulile i)-iv).

6. Să se determine formula termenului general al şirului $(X_n)_n \in N$ care satisface recurenţa:

$X_{n+2}=X_{n+1}+X_n+3n+2$, X_0, X_i daţi.

7. Fie $(a_i)_{i=1,n}$ un şir de numere reale şi $P(x)=\prod_{i=1}^{n}(x-a_i) = \sum_{k=0}^{n} b_k x^k$. Să se scrie un algoritm care calculează coeficienţii $(b_k)_{k=1,n}$.

Indicaţie. Se vor considera polinoamele P(x)= $\prod_{i=1}^{n}(x-a_i)$, k= $\overline{1,n}$ şi se va observa ce relaţie există între coeficienţii lui P_k şi P_{k+1}. În plus, se va observa că :

$P_{k+1}(x)=(x-a_{k+1})\ P_k$

8. Considerăm următorul algoritm de generare a numerelor aleatoare: se porneşte cu un număr de două cifre $X_0=\overline{AB}$ şi se ridică la pătrat, obţinându-se numărul de patru cifre Y= $\overline{CDEF}$ din care se extrage următorul număr aleatoriu $X_1=\overline{DE}$ şi se repetă procedeul cu el.

i) Să se demonstreze că şirul $(X_n)_{n\in N}$ este periodic de perioadă T_{X^0} oricare ar fi $X_0 \in \overline{1,99}$

ii) Să se determine numărul X_0 astfel încât şirul $(X_n)_{n\in N}$ să aibă perioada maximă, respectiv minimă.

iii) Să se calculeze funcţia de operaţii pentru algoritmul de la punctul ii).

9. Se dă funcţia F: N $\rightarrow$ N dată de relaţia

$$F(x)=\begin{cases}2F\left(\dfrac{x-1}{2}\right)+1 \text{ dacă x impar}\\ 2F(F\left(\dfrac{x}{2}\right)), \text{ dacă x par, } x\neq 0\\ 0 \quad \text{altfel}\end{cases}$$

Demonstraţi că expresia lui F(x) este corect definită, i.e. calculul lui F(x) se termină într-un număr finit de paşi oricare ar fi x $\in$ N.

10. Să considerăm problema evaluării unui polinom într-un punct. Fiind daţi n coeficienţi, a_0,a_1, …, a_{n-1} şi un număr real x, dorim să calculăm $\sum_{i=0}^{n-1}a_i x^i$. Descrieţi un algoritm simplu în timp $O(n^2)$ pentru această operaţie. Descrieţi şi un algoritm în timp O(n) care foloseşte următoarea metodă de rescriere a unui polinom, numită schema lui Horner:

$$\sum_{i=0}^{n-1}a_i x_i = (...(a_{n-1}x+a_{n-2})x+...+a_1)x+a_0$$

11. Descrieţi un algoritm al cărui timp de execuţie să fie O(*n lg n*) şi care, pornind de la o mulţime dată S de *n* numere reale şi un alt număr real x, să decidă dacă printre elementele lui S există două elemente având suma *x*.

12. Calculaţi timpul de execuţie al următoarelor funcţii:

a) $f(n)=3n^2-n+4$

b) $f(n)=\sum_{i=0}^{n} i$

c) *f(n)+g(n),* unde *f(n)=4n*^{2}*-2n+3, g(n)=log n*

d) *f(n)+g(n),* unde *f(n)=* $\sqrt{n}$ *, g(n)=n log n+5*

13. Numerele lui Fibonacci de ordin k>=2 sunt date de:

$$F_n^{(k)} = \begin{cases} 0 & 0 \le n < k-1 \\ 1 & n = k-1 \\ \sum_{i=1}^{n} F_{n-i}^{(k)} & n \ge k \end{cases}$$

Scrieţi variantele recursive şi nerecursive ale funcţiilor care calculează $F_n^{(k-1)}$.

14. Scrieţi câte o variantă recursivă pentru metodele coardei şi a înjumătăţirii intervalului şi comparaţi funcţiile lor de operaţii cu cele ale variantelor iterative.

15. Să se rezolve următoarele recurenţe:

a) T(1)=8 şi T(n)=3T(n-1)-15, pentru n>1
b) T(1)=2 şi T(n)=T(n-1)+n-1, pentru n>1
c) T(1)=3 şi T(n)=T(n-1)+2n-3, pentru n>1
d) T(1)=1 şi T(n)=2T(n-1)+n-1, pentru n>1
e) T(1)=5 şi T(n)=2T(n-1)+3n+1, pentru n>1
f) T(1)=3 şi T(n)=4T(n/3)+2n-1, pentru n>1
g) T(1)=1 şi T(n)=2T(n/2)+6n-1, pentru n>1

16. Scrieţi un algoritm recursiv şi unul iterativ pentru calculul factorialului şi comparaţi funcţiile de operaţii în cele două cazuri.