

Concurs de admitere, Varianta 1
Facultatea de Matematică și Informatică
Universitatea de Vest din Timișoara

Sesiunea Septembrie, 2024

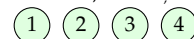
-
1. Timp total de lucru: 3 ore
 2. Toate subiectele sunt obligatorii!
 3. Citiți cu atenție informațiile legate de tipurile de întrebări, prezentate mai jos.
-

Tipuri de întrebări

Pe foaia de concurs veți întâlni trei tipuri de întrebări. Pentru a le recunoaște mai ușor, în partea dreaptă veți regăsi un șablon pentru răspunsuri, cu indicații legate de tipul de întrebare.

Tipul 1 Pentru întrebările de acest tip veți marca pe foaia de concurs un singur răspuns corect. Le veți recunoaște după modelul indicat în partea dreaptă a acestui rând.

MODEL, marcați exact un răspuns:



Punctaj pe întrebare: 2p (standard)+1p (bonus pentru răspuns complet).

Tipul 2 Pentru întrebările de acest tip veți marca pe foaia de concurs fie un singur răspuns corect, fie două răspunsuri corecte. Le veți recunoaște după modelul indicat în partea dreaptă a acestui rând.

MODEL, marcați 1-2 răspunsuri:



Punctaj maxim pe întrebare: 2p (standard)+1p (bonus pentru răspuns complet) sau 2x2p (standard, pe variantă)+1p (bonus pentru răspuns complet).

Tipul 3 Pentru întrebările de acest tip veți marca pe foaia de concurs fie un singur răspuns corect, fie două răspunsuri corecte, fie trei răspunsuri corecte. Le veți recunoaște după modelul indicat în partea dreaptă a acestui rând.

MODEL, marcați 1-3 răspunsuri:



Punctaj maxim pe întrebare: 2p (standard)+1p (bonus pentru răspuns complet) sau 2x2p/2x3p (standard, pe variantă)+1p (bonus pentru răspuns complet).

Punctaj

Fiecare variantă de răspuns corectă valorează exact 2 puncte. Dacă răspunsul oferit identifică toate variantele corecte de răspuns, la punctajul întrebării se adaugă un bonus pentru răspuns complet de 1 punct. Orice răspuns greșit anulează întregul punctaj al întrebării respective, indiferent de numărul de răspunsuri corecte.

La punctajul obținut se adaugă 10 puncte, pentru start.

Pentru nota 5 sunt necesare 49 de puncte, inclusiv punctele de start, pentru nota 10 sunt necesare cel puțin 98 de puncte, inclusiv punctele de start.

Partea I

Problema 1: Factorial sau nu?

Se consideră funcțiile F_1 și F_2 definite în modul ilustrat mai jos:

```
funcția  $F_1$ (întreg  $n$ ) : întreg
    dacă  $n == 0$  atunci
        întoarce 1
    altfel
        întoarce  $n * F_1(n-1)$ 
```

```
funcția  $F_2$ (întreg  $n$ ) : întreg
    dacă  $n != 1$  atunci
        întoarce  $n * F_2(n-1)$ 
    altfel
        întoarce 0
```

Care din funcțiile F_1 sau F_2 returnează 120 când este apelată cu parametrul de intrare $n = 5$?

Ciornă, marcați exact un răspuns:

☐ 1 ☐ 2 ☐ 3 ☐ 4

- ☐ 1 atât F_1 cât și F_2 ☒ 2 numai F_1 ✓ ☐ 3 numai F_2 ☐ 4 nici F_1 nici F_2

Explicații

Se observă ușor că, dacă $n \geq 0$ atunci $F_1(n)$ returnează valoarea lui $n!$, iar $F_2(n)$ returnează 0. Întrucât $120 = 5!$, rezultă că răspunsul corect este "numai F_1 ".

Problema 2: Grafuri parțiale

Determinați numărul grafurilor parțiale distincte cu număr impar de arce pentru graful orientat dat prin matricea de adiacență alăturată:

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Ciornă, marcați exact un răspuns:

☐ 1 ☐ 2 ☐ 3 ☐ 4

- ☐ 1 8 ☒ 2 128 ✓ ☐ 3 120 ☐ 4 2

Explicații

Graful descris cu această matrice de adiacență are numărul de arce egal cu numărul de elemente 1 din matrice, adică 8. Numărul grafurilor parțiale distincte cu număr impar de arce este $N_1 + N_3 + N_5 + N_7$, unde fiecare N_k este numărul grafurilor parțiale distincte cu k arce. Un graf parțial cu k arce se obține selectând k arce din cele 8 arce ale grafului nostru. Deci $N_k = C_8^k$.

Prin urmare, numărul căutat este $C_8^1 + C_8^3 + C_8^5 + C_8^7 = 9 + 56 + 56 + 8 = 128$.

Problema 3: Mister dezlegat

Într-o pungă sunt 10 jetoane numerotate de la 1 la 10. Anca, Bianca, Călin, Daniela și Eugen extrag câte două jetoane din pungă, fără să le arate celorlalți numerele de pe acestea. Între aceste persoane are loc următoarea discuție:

Anca: - Suma numerelor de pe jetoanele mele este 16.

Bianca: - Suma numerelor de pe jetoanele mele este 11.

Călin: - Suma numerelor de pe jetoanele mele este 17.

Daniela: - Eu nu vă spun nimic despre jetoanele mele.

Eugen, care a extras jetoanele cu numerele 1 și 3, spune, după un timp de gândire:

- Nu-i nimic, vă spun eu care sunt numerele de pe jetoanele Danielei.

Care sunt numerele de pe jetoanele extrase de Daniela?

Ciornă, marcați exact un răspuns:

① ② ③ ④

- ① 4,6. ② 5,9. ③ 2,5. ✓ ④ 2,6

Explicații

Fie x_A, y_A numerele Ancăi, x_B, y_B numerele Biancăi, x_C, y_C numerele lui Călin, și x_D, y_D numerele Danielei. Presupunem că $x_A < y_A$, $x_B < y_B$, $x_C < y_C$ și $x_D < y_D$.

Știm că Eugen a extras jetoanele 1 și 3, deci:

1. $x_A, y_A, x_B, y_B, x_C, y_C, x_D, y_D$ sunt numere distincte din mulțimea $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} - \{1, 3\} = \{2, 4, 5, 6, 7, 8, 9, 10\}$.

2. $(x_A, y_A) \in \{(6, 10), (7, 9)\}$ pentru că $x_A + y_A = 16$.

3. $(x_C, y_C) \in \{(7, 10), (8, 9)\}$ pentru că $x_C + y_C = 17$.

Singura posibilitate să satisfacem simultan 2 și 3 este $(x_A, y_A) = (6, 10)$ și $(x_C, y_C) = (8, 9)$. Deducem că $x_B, y_B, x_D, y_D \in \{2, 4, 5, 7\}$ și că

4. $(x_B, y_B) = (4, 7)$ pentru că $x_B + y_B = 11$.

Prin urmare $(x_D, y_D) = (2, 5)$.

Problema 4: Căutare binară

Fie tabloul unidimensional cu elementele $(20, 15, 12, 8, 4, 2, 1)$, în această ordine. Pentru a verifica dacă numărul 5 se află printre elementele tabloului, se aplică metoda căutării binare.

Care este succesiunea corectă de elemente cu care se compară numărul căutat?

Ciornă, marcați exact un răspuns:

① ② ③ ④

- ① 20, 15, 12, 8, 4. ③ 20, 12, 4.
② 8, 4, 2. ④ 8, 2, 4. ✓

Explicații

Căutarea binară se poate face într-un vector v numai dacă elementele acestuia sunt ordonate.

În cazul nostru, elementele sunt ordonate descrescător, prin urmare pseudocodul căutării binare este

procedura CAUTAREBINARA(întreg x , întreg $v[1..n]$)

 întreg $st = 1, dr = n, poz = 0, m$

cât timp $st \leq dr$ **ȘI** $poz == 0$ **execută**

$m = (st + dr) \% 2$

dacă $v[m] == x$ **atunci**

$poz = m$

altfel

dacă $v[m] > x$ **atunci**

$st = m + 1$

altfel

$dr = m - 1$

dacă $poz \neq 0$ **atunci**

x apare în v la poziția poz

altfel

x nu apare în v

Vectorul $v = (20, 15, 12, 8, 4, 2, 1)$ are lungimea $n = 7$ iar căutarea binară îl instanțiază pe m cu valorile 4, 6, 5, în această ordine. Prin urmare, $x = 5$ este comparat cu elementele $v[4], v[6], v[5]$, adică cu 8, 2, 4 în această ordine.

Problema 5: Cuvinte speciale

Utilizând metoda backtracking se generează în ordine lexicografică cuvintele de câte patru litere din mulțimea $A = \{a, b, c, d, e\}$, cuvinte care nu conțin două vocale alăturate. Primele opt cuvinte generate sunt, în ordine: abab, abac, abad, abba, abbb, abbc, abbd, abbe.

Câte dintre cuvintele generate încep cu litera b și se termină cu e?

Ciornă, marcați exact un răspuns:

☐ 1 ☐ 2 ☐ 3 ☐ 4

☐ 0 ☒ 15 ☐ 12 ☐ 20

Explicații

Cuvintele de acest fel sunt de forma $bxye$ cu $x \in A$ și y consoană. x este una din 5 litere posibile iar y este una din 3 consoane posibile. Rezultă că numărul de astfel de șiruri este $5 \cdot 3 = 15$.

Problema 6: Ce face?

Se consideră algoritmul alăturat, reprezentat în pseudocod:

 citește n (număr natural)

```

m = 0
p = 1
cât timp n > 0 execută
    dacă n%2 ≠ 0 atunci
        n = n - 1
        m = m + (n%10) * p
        n = [n/10]
        p = 10 * p
scrie m
    
```

S-a notat cu $x \% y$ restul împărțirii numărului natural x la numărul natural nenul y și cu $[z]$ partea întreagă a numărului real z .

Câte numere naturale, fiecare având exact patru cifre, pot fi citite pentru variabila n astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea să se afișeze valoarea 40?

Ciornă, marcați exact un răspuns:

- ① ② ③ ④

- ① Nici unul. ② 6 ③ 8 ✓ ④ 24

Explicații

Pentru orice număr natural cu reprezentarea zecimală $a_1a_2 \dots a_n$, programul afișează numărul cu reprezentarea zecimală $b_1b_2 \dots b_n$ unde fiecare b_i este cea mai mare cifră pară astfel încât $b_i \leq a_i$.

Dacă numărul afișat este $m = 40$ atunci reprezentarea zecimală a lui n este de forma $1xyz$ unde $x \in \{0, 1\}$, $y \in \{4, 5\}$, și $z \in \{0, 1\}$. Sunt $2^3 = 8$ astfel de numere.

Problema 7: Operații aritmetice

Se consideră funcția F descrisă în pseudocod

```

funcția F(întreg x, întreg y)
    întreg acc = 0
    cât timp x ≠ 0 execută
        dacă x%2 > 0 atunci
            acc = acc + y
            x = x/2
            y = y * 2
        întoarce acc
    
```

S-a notat cu $x \% y$ restul împărțirii numărului natural x la numărul natural nenul y , și cu x/y câtul împărțirii numărului natural x la numărul natural nenul y . Care este cel mai mare număr prim y astfel încât $F(x, y)$ să returneze 231?

Ciornă, marcați exact un răspuns:

- ① ② ③ ④

- ① 3 ② 7 ③ 11 ✓ ④ 21

Explicații

Se observă că dacă x, y sunt numere naturale atunci $F(x, y)$ calculează produsul $x \cdot y$ în timp logaritm față de x . Întrucât descompunerea în factori primi a lui 231 este $231 = 3 \cdot 7 \cdot 11$, răspunsul este 11: valoarea 231 este returnată de apelul $F(21, 11)$.

Partea a II-a

Problema 8: O matrice simetrică

Se știe că a este un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 1 la 5, având inițial toate elementele nule. Notăm cu $x \% y$ restul împărțirii numărului natural x la numărul natural nenul y .

Pentru ca algoritmul următor să instanțieze elementele tabloului a cu valorile din tabloul alăturat condițiile (A) și (B) trebuie înlocuite în felul următor

Ciornă, marcați 1-2 răspunsuri:

procedura INSTANȚIAZĂ()

 întreg i, j

pentru $i = 1$ **to** 5 **execută**

pentru $j = 1$ **to** 5 **execută**

dacă (A) **atunci** (B)

0	3	0	5	0
3	0	5	0	7
0	5	0	7	0
5	0	7	0	9
0	7	0	9	0

① ② ③ ④ ⑤

① (A) se înlocuiește cu $(i + j) \% 2 == 0$

② (A) se înlocuiește cu $(i + j) \% 2 > 0$ ✓

③ (A) se înlocuiește cu $(i - j) \% 2 == 0$

④ (B) se înlocuiește cu $a[i][j] = i + j$ ✓

⑤ (B) se înlocuiește cu $a[i][j] = 2 * i + 1$

Explicații

Se observă că tabloul a este instanțiat astfel încât $a[i][j] = i + j$ pentru toți $1 \leq i, j \leq n$ care satisfac condiția că $i + j$ este număr impar, adică $(i + j) \% 2 > 0$.

Problema 9: Trunchieri

Definim operația de trunchiere a unui număr natural cu reprezentarea în baza 10, $\overline{c_1 c_2 \dots c_k}$, ca fiind

$$\text{trunc}(\overline{c_1 c_2 \dots c_k}) = \begin{cases} 0 & \text{dacă } k < 2, \\ \overline{c_1 c_2} & \text{în caz contrar.} \end{cases}$$

Presupunem că x este un tablou de n numere naturale mai mici decât 1000000, și că operația $\text{div}(a, b)$ returnează câtul împărțirii numărului natural a la numărul natural nenul b .

Precizați care dintre următoarele funcții calculează suma trunchierilor elementelor lui x . De exemplu, dacă $n = 4$ și $x = \{213, 7, 78347, 22\}$ atunci suma trunchierilor este $21 + 0 + 78 + 22 = 121$.

Ciornă, marcați 1-2 răspunsuri:

① Niciuna dintre funcțiile de mai jos

① ② ③ ④ ⑤

② ✓

```

funcția T1(întreg  $n$ , întreg  $x[1..n]$ ): întreg
    întreg  $s = 0, m = n$ 
    cât timp  $m > 0$  execută
        dacă  $x[m] > 9$  atunci
            cât timp  $x[m] > 99$  execută
                 $x[m] = \text{div}(x[m], 10)$ 
                 $s = s + x[m]$ 
             $m = m - 1$ 
    întoarce  $s$ 
    
```

③

```

funcția T2(întreg  $n$ , întreg  $x[1..n]$ ): întreg
    întreg  $s = n, m = n$ 
    cât timp  $m > 0$  execută
        dacă  $x[m] > 9$  atunci
            cât timp  $x[m] > 99$  execută
                 $x[m] = \text{div}(x[m], 10)$ 
                 $s = s + x[m]$ 
             $m = m - 1$ 
    întoarce  $s$ 
    
```

④

```

funcția T3(întreg  $n$ , întreg  $x[1..n]$ ): întreg
    întreg  $s = 0, m = n$ 
    cât timp  $m > 0$  execută
        dacă  $x[m] > 9$  atunci
            cât timp  $x[m] > 99$  execută
                 $x[m] = \text{div}(x[m], 10)$ 
                 $s = s + x[m]$ 
             $m = m - 1$ 
    întoarce  $s$ 
    
```

⑤ ✓

```

funcția T4(întreg  $n$ , întreg  $x[1..n]$ ): întreg
    întreg  $s = 0, m = 1$ 
    cât timp  $m \leq n$  execută
        dacă  $x[m] > 9$  atunci
            cât timp  $x[m] > 99$  execută
                 $x[m] = \text{div}(x[m], 10)$ 
                 $s = s + x[m]$ 
             $m = m + 1$ 
    întoarce  $s$ 
    
```

Explicații

T1(n, x) însumează trunchierile lui $x[n], \dots, x[1]$ în această ordine.

T4(n, x) însumează trunchierile lui $x[1], \dots, x[n]$ în această ordine.

Problema 10: Swap

Se știe că n este un număr întreg mai mare ca 1, și că v este un vector de n numere indexate de la 1 la n . Care dintre următoarele afirmații referitoare la tabloul v sunt adevărate după rularea algoritmului PARCURRE(), oricare ar fi valorile stocate în tabloul v ?

Alegeți toate variantele corecte.

Ciornă, marcați 1-2 răspunsuri:

① ② ③ ④ ⑤

procedura PARCURRE()

```

    întreg  $tmp, i = 1$ 
    pentru  $i = 1, n - 1$  execută
        dacă  $(v[i] < v[i + 1])$  atunci
             $tmp = v[i]$ 
             $v[i] = v[i + 1]$ 
             $v[i + 1] = tmp$ 
    
```

① $v[1] < v[i]$ pentru toți $1 < i \leq n$.

② $v[n] \leq v[i]$ pentru toți $1 \leq i < n$. ✓

③ $v[n - 1] \geq v[n]$. ✓

④ $v[i] \geq v[i + 1]$ pentru toți $1 \leq i \leq n$.

⑤ $v[1] \geq v[2]$.

<pre> ok = 1 altfel x = x - 1 dacă (ok == 1) atunci afișează x altfel afișează "Nu exista" procedura CAUTAB(întreg v[1..n], întreg n) întreg i, x, ok = 0 întreg f[1..999] pentru i = 1, n execută f[i] = 0 pentru i = 1, n execută </pre>	<pre> dacă v[i] < 1000 atunci f[i] = f[i] + 1 x = 999 cât timp (x ≥ 100) ȘI (ok == 0) execută dacă (f[x] == 0) ȘI (f[1000 - x] ≠ 0) atunci ok = 1 altfel x = x - 1 dacă (ok == 1) atunci afișează x altfel afișează "Nu exista" </pre>
--	---

Care dintre următoarele afirmații referitoare la aceste proceduri sunt adevărate? Alegeți toate variantele corecte.

Ciornă, marcați 1-2 răspunsuri:

☐ 1
 ☐ 2
 ☐ 3
 ☐ 4
 ☐ 5

- | | |
|---|--|
| <p>① Doar procedura CAUTAA(v) este corectă. ✓</p> <p>② Ambele proceduri sunt corecte.</p> <p>③ Doar procedura CAUTAB(v) este corectă.</p> | <p>④ CAUTAA(v) are ordinul de complexitate $O(\log n)$.</p> <p>⑤ CAUTAA(v) are ordinul de complexitate $O(n)$. ✓</p> |
|---|--|

Explicații

CAUTAA(v) instanțiază fiecare element $f[i]$ al lui f cu o valoare nenulă dacă și numai dacă i apare în v .

Prin urmare ok devine 1 dacă și numai dacă există $100 \leq k \leq 999$ astfel încât x nu apare în v și $1000 - x$ apare în v . Deci procedura CAUTAA(v) este corectă.

Procedura CAUTAB(v) este incorectă pentru că, dacă elementele lui v sunt (499, 501) atunci căutarea ar trebui să raporteze "nu exista", dar CAUTAB(v) găsește valoarea incorectă $x = 999$:

- se instanțiază $f[1] = f[2] = 1$ și $f[i] = 0$ pentru toți $3 \leq i \leq 1000$.
- ok devine 1 când $x = 999$ pentru că $f[999] = 0$ și $f[1000 - 999] = f[1] = 1 > 0$.

Valoarea 999 este incorectă fiindcă egalitatea $999 + b = 1000$ nu are loc pentru nici un element b din v .

Ordinul de complexitate al lui CAUTAA(v) este $O(n)$.

Problema 13: Rezultat simplu

Se consideră funcția RUNME, de mai jos

Cerințe: $k \in \mathbb{N}$

```

funcția RUNME(k)
    i, rezultat = 0
    pentru i = 0 to k execută
    
```

Ciornă, marcați 1-2 răspunsuri:

☐ 1
 ☐ 2
 ☐ 3
 ☐ 4
 ☐ 5

```

    |   dacă  $i \bmod 3 == 1$  atunci
    |   |   rezultat = rezultat + i
    |   altfel
    |   |   rezultat = rezultat + 1
    |   întoarce rezultat

```

Precizați care dintre următoarele afirmații sunt adevărate:

- ① Pentru apelul $\text{RUNME}(5)$, funcția va returna 8 ④ Pentru apelul $\text{RUNME}(9)$, funcția va returna 18
 ② Pentru apelul $\text{RUNME}(0)$, funcția va returna 1 ✓ ⑤ Pentru apelul $\text{RUNME}(4)$, funcția va returna 8 ✓
 ③ Pentru apelul $\text{RUNME}(5)$, funcția va returna 5

Problema 14: Zigzag

Robotul Robi poate fi ghidat să execute una din următoarele 4 comenzi de a înainta 1 metru: N spre nord, V spre vest, E spre est, și S spre sud. Din cauza unui defect de fabricație, Robi nu poate efectua aceeași comandă succesiv și să meargă mai mult de 1 metru în aceeași direcție. De pildă, secvența NVN îl deplasează pe Robi 2 metri spre nord și 1 metru spre vest, dar secvența NNV nu poate fi executată fiindcă comanda N apare consecutiv. Alex, elev în clasa a XII-a, este preocupat să rezolve două probleme:

1. Pentru orice valori întregi x, y , să găsească o secvență cât mai scurtă de comenzi $\pi_{x,y}$ care să-l ghideze pe Robi să ajungă x metri mai la sud și y metri mai la est.
2. Să găsească o formulă de calcul pentru numărul s_n de secvențe de comenzi care-l ghidează pe Robi să se deplaseze n metri.

Pentru găsirea lui $\pi_{x,y}$, Alex folosește metoda de parcurgere în lățime respectând următoarele reguli:

- (1) vizitarea locurilor se face în ordinea N,E,S,V; și
- (2) se evită secvențe de comenzi ce provoacă *cicluri*, adică vizitarea aceluiași loc de mai multe ori.

Care dintre următoarele afirmații sunt adevărate? Alegeți toate variantele corecte.

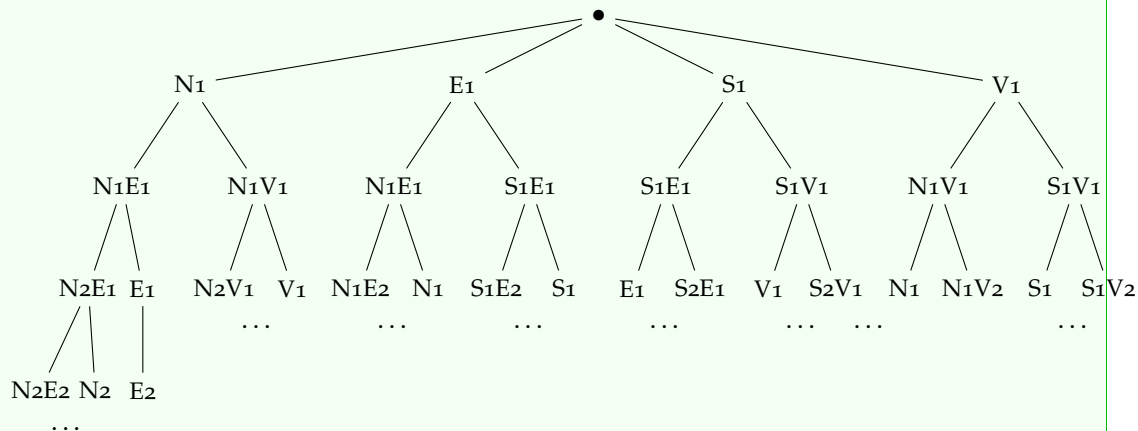
Ciornă, marcați 1-2 răspunsuri:

① ② ③ ④ ⑤

- ① $\pi_{2,1}$ este o secvență de 3 comenzi. ✓ ④ $s_n = 4 \cdot 3^{n-1}$ pentru toți $n \geq 1$. ✓
 ② În general $\pi_{x,y}$ este o secvență de $|x| + |y|$ comenzi.
 ③ $s_n = 2^{n-1}$ pentru toți $n \geq 1$. ⑤ Ultima comandă din $\pi_{0,2}$ este N.

Explicații

Fiecare comandă îl ghidează pe robot să avanseze 1 metru. Deci secvențele de comenzi care îl ghidează să parcurgă n metri sunt cele de forma $c_1c_2 \dots c_n$, unde $c_1 \in \{N, E, S, V\}$ și $c_i \in \{N, E, S, V\} - \{c_{i-1}\}$ pentru toți $1 < i \leq n$. Deci $s_n = 4 \cdot 3^{n-1}$ pentru toți $n \geq 1$. Arborele de căutare pentru parcurgerea în lățime a lui Robi este



de unde obținem secvența $\pi_{2,1} = \text{SES}$ pentru calea la nodul $S2E1$, și secvența $\pi_{0,2} = \text{NESE}$ pentru calea la nodul $E2$. Deci

- $\pi_{2,1}$ este o secvență de 3 comenzi
- Afirmația că $\pi_{x,y}$ are $|x| + |y|$ comenzi este falsă fiindcă $\pi_{0,2} = \text{NESE}$ are 4 comenzi, și $4 \neq |0| + |2|$
- Ultima comandă din $\pi_{0,2}$ nu este N.

Problema 15: Relații de prietenie

Se consideră o matrice de adiacență, M , care reprezintă prietenii dintre n persoane (unde $M[i][j] = 1$ înseamnă că persoana i este prietenă cu persoana j , iar $M[i][j] = 0$ înseamnă că nu sunt prieteni, iar $M[i][i] = 1, \forall i \in \{1, 2, \dots, n\}$).

Care dintre următoarele metode determină corect dacă există cel puțin două persoane cu același număr de prieteni?

Ciornă, marcați 1-2 răspunsuri:

① ② ③ ④ ⑤

- ① Pentru fiecare persoană i se calculează numărul de prieteni, N_i , adunând valorile de pe rândul i al matricei și stocând aceste valori într-un tablou. Dacă tabloul conține cel puțin două valori identice, se returnează True; altfel, se returnează False. ✓
- ② Pentru fiecare persoană i din grup, se calculează numărul de prieteni adunând valorile din rândul i al matricei. Dacă numărul de prieteni este mai mare decât $\frac{n}{2}$, se returnează True, iar altfel, False.
- ③ Pentru fiecare persoană i se calculează numărul de prieteni, N_i , adunând valorile de pe rândul i al matricei și se incrementează valoarea $A[N_i]$, unde A este un vector de frecvență inițializat cu 0. Dacă $A[N_i] = 2$ se va returna True. Dacă se termină execuția fără a întâlni acest caz, se returnează False. ✓
- ④ Pentru fiecare persoană i din grup, se calculează numărul de prieteni adunând valorile de pe rândul i al matricei. Dacă vreo persoană are 0 prieteni, se returnează True, altfel se returnează False.

- ⑤ Pentru fiecare persoană i din grup, se calculează numărul de prieteni adunând valorile de pe rândul i al matricei. Dacă vreo persoană are 0 prieteni, se returnează False, altfel se returnează True.

Explicații

Răspunsul 1 va calcula numărul de prieteni pentru fiecare persoană iar apoi verifică dacă există două persoane cu același număr de prieteni.

Răspunsul 3 va calcula numărul de prieteni, N_i pentru o persoană și verifică dacă acest N_i a mai fost întâlnit.

Răspunsurile 2,4,5 nu garantează că există două persoane cu același număr de prieteni

Partea a III-a

Problema 16: Pune 'T'

Se consideră o matrice pătratică mat de dimensiune $n \times n$ (n număr impar, $3 \leq n \leq 100$) și subalgoritmul $PUNET(mat, n, i, j)$ care pune caracterul 'T' pe anumite poziții în matricea mat . Parametrii i, j sunt numere naturale ($1 \leq i \leq n$ și $1 \leq j \leq n$), iar pseudocodul subalgoritmului este

```

procedura PUNET(char  $mat[1..n][1..n]$ , întreg  $n$ , întreg  $i$ , întreg  $j$ )
    dacă  $i \leq div(n, 2)$  atunci
        dacă  $j \leq n - i$  atunci
             $mat[i][j] = 'T'$ 
            PUNET( $mat, n, i, j + 1$ )
        altfel
            PUNET( $mat, n, i + 1, i + 2$ )

```

unde $div(n, 2)$ returnează câtul împărțirii întregi a lui n la 2.

Precizați de câte ori se autoapelează subalgoritmul $PUNET(mat, n, i, j)$ dacă se evaluează $PUNET(mat, 7, 2, 4)$ (Apelul inițial nu se ia în considerare). Alegeți toate variantele corecte.

Ciornă, marcați 1-3 răspunsuri:

① ② ③ ④ ⑤

- ① de 5 ori. ✓ ③ de o infinitate de ori.
 ② de 12 ori. ④ de 0 ori.

- ⑤ de același număr de ori ca și în cazul apelului $PUNET(mat, 9, 3, 5)$.
 ✓

Explicații

Secvența de auto-apeluri ale lui $\text{PUNET}(mat, 7, 2, 4)$ este

$\rightarrow \text{PUNET}(mat, 7, 2, 5) \rightarrow \text{PUNET}(mat, 7, 2, 6) \rightarrow \text{PUNET}(mat, 7, 3, 4) \rightarrow \text{PUNET}(7, 3, 5) \rightarrow \text{PUNET}(7, 4, 5)$
iar secvența de auto-apeluri ale lui $\text{PUNET}(mat, 9, 3, 5)$ este

$\rightarrow \text{PUNET}(mat, 9, 3, 6) \rightarrow \text{PUNET}(mat, 9, 3, 7) \rightarrow \text{PUNET}(mat, 9, 4, 5) \rightarrow \text{PUNET}(9, 4, 6) \rightarrow \text{PUNET}(9, 5, 6)$

Deci numărul de autoapeluri ale lui $\text{PUNET}(mat, n, i, j)$ dacă se evaluează

$\text{PUNET}(mat, 7, 2, 4)$ este 5,

care este același dacă se evaluează $\text{PUNET}(mat, 9, 3, 5)$.

Problema 17: Șiruri magice

Presupunem că A este un alfabet de $n > 1$ litere. Spunem că s este un *șir magic* dacă s are lungimea $n + 2$ și fiecare literă din A apare cel puțin o dată în s . Fie x_n numărul șirurilor magice cu litere din alfabetul A .

Care dintre următoarele afirmații referitoare la x_n sunt adevărate?
Alegeți toate variantele corecte.

Ciornă, marcați 1-3 răspunsuri:

- ① ② ③ ④ ⑤

- ① $x_n = \frac{n! \cdot (n+2) \cdot (3n+1)}{24}$ ✓ ③ $x_n = (n+1)! + n!$
② $x_n = \frac{(n+2)! \cdot (3n+1)}{4! \cdot (n+1)}$ ✓ ④ $x_2 = 14$ ✓
⑤ $x_3 = 30$

Explicații

Fie s un șir magic. Distingem 2 cazuri distincte:

1. s are 2 litere L_1, L_2 distincte care apar de câte 2 ori în s , iar restul literelor din A apar o singură dată.

- Sunt C_n^2 posibilități să alegem $\{L_1, L_2\}$ din alfabetul A .
- Sunt C_{n+2}^2 posibilități să alegem pozițiile unde apare L_1 în șirul s .
- Sunt C_n^2 posibilități rămase să alegem pozițiile unde apare L_2 în s .
- Sunt $(n-2)!$ posibilități să aranjăm literele lui $A - \{L_1, L_2\}$ pe cele $n-2$ poziții rămase din s .

Conform regulii produsului, avem $C_n^2 \cdot C_{n+2}^2 \cdot C_n^2 \cdot (n-2)! = C_n^2 \cdot \frac{(n+2)!}{4}$ șiruri magice în acest caz.

2. s are o literă L care apare de 3 ori în s , iar restul literelor din A apar o singură dată.

- Sunt n posibilități de a alege litera L din A .
- Sunt C_{n+2}^3 moduri să alegem pozițiile unde apare L în s .
- Sunt $(n-1)!$ posibilități să aranjăm literele lui $A - \{L\}$ pe cele $n-1$ poziții rămase din s .

Conform regulii produsului, avem $n \cdot C_{n+2}^3 \cdot (n-1)! = n \cdot \frac{(n+2)!}{6}$ șiruri magice în acest caz.

Deducem că

$$x_n = C_n^2 \cdot \frac{(n+2)!}{4} + n \cdot \frac{(n+2)!}{6} = \frac{(n+2)!n(3n+1)}{24}.$$

Problema 18: O funcție recursivă

Se consideră funcția recursivă de mai jos:

funcția f (întreg n): întreg

dacă $n \leq 3$ **atunci**

întoarce 1

altfel

întoarce $f(n-1) + 2 \cdot f(n-3)$

Dacă primul apel este $f(10)$, de câte ori se calculează $f(4)$? Alegeți toate variantele corecte.

Ciornă, marcați 1-3 răspunsuri:

① ② ③ ④ ⑤

- ① De mai puțin de 26 ori ✓ ④ Niciodată
- ② De mai mult de 10 ori
- ③ De 6 ori ✓ ⑤ O dată

Explicații

Fie $x(n)$ numărul de calcule ale lui $F(4)$ pe durata unui apel $F(n)$. Se observă că $x(n) = x(n-1) + x(n-2) + x(n-3)$ dacă $n \geq 5$ și că $x(4) = 1$, $x(3) = x(2) = x(1) = 0$. Deci $x(5) = x(4) + x(2) = 1$, $x(6) = x(5) + x(3) = 1$, $x(7) = x(6) + x(4) = 2$, $x(8) = x(7) + x(5) = 2 + 1 = 3$, $x(9) = x(8) + x(6) = 3 + 1 = 4$, $x(10) = x(9) + x(7) = 4 + 2 = 6$. Așadar $F(4)$ se calculează de exact 6 ori.

Problema 19: Găsește arborele

Presupunem că T este un arbore cu 6 noduri, numerotate de la 1 la 6, în care nodul 1 este rădăcină. Deaseme-nea, presupunem că V este vectorul de „tați” ai nodurilor din T , adică $V[i]$ este nodul „tată” al lui i pentru toți $1 < i \leq 6$. Se consideră următoarea operație prin care se transformă arborele T : se elimină din arbore nodul „frunză” numerotat cu valoarea minimă, marcându-se nodul „tată” al acestuia. Se știe că dacă asupra arborelui T se efectuează de patru ori succesiv operația de transformare menționată mai sus, atunci se marchează, în această ordine, nodurile 5, 1, 1, 1.

Care dintre următoarele afirmații referitoare la arborele T sunt adevărate?

Ciornă, marcați 1-3 răspunsuri:

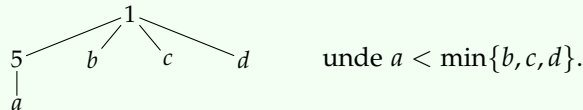
- ① ② ③ ④ ⑤

Alegeți toate variantele corecte.

- ① T are 3 noduri „frunză”. ④ $V[6] = 1$. ✓
 ② T are 4 noduri „frunză”. ✓
 ③ $V[5] = 6$. ⑤ $V[4] = 1$. ✓

Explicații

Din faptul că transformarea menționată marchează nodurile în ordinea 5, 1, 1, 1 rezultă că T este



Rezultă că $a = 2$ și $\{b, c, d\} = \{3, 4, 6\}$, deci $V[2] = 5$, $V[3] = V[4] = V[5] = V[6] = 1$. Arborele T are 4 noduri „frunză”: 2, 3, 4, 6.

Problema 20: Detecția elementelor minimale ale unei matrici

Presupunem că a este un tablou bidimensional cu m linii și n coloane numerotate începând de la 1, ale cărui elemente sunt numere întregi. Programul următor trebuie să ia ca argumente de intrare numerele m, n și tabloul a , și să afișeze pozițiile elementelor din tablou care sunt strict mai mici decât toate elementele cu care se învecinează direct (aflate pe aceeași linie dar pe o coloană alăturată sau pe aceeași coloană dar pe o linie alăturată).

procedura POZITIIIMINIMALE(întreg m , întreg n , întreg $a[1..m][1..n]$)

 întreg i, j, b

```

pentru  $i = 1, m$  execută
  pentru  $j = 1, n$  execută
     $b = 1$ 
    dacă  $((i - 1 \geq 1) \text{ ŞI } (A)) \text{ SAU } ((i + 1 \leq n) \text{ ŞI } (B))$  atunci
       $b = 0$ 
    dacă  $((j - 1 \geq 1) \text{ ŞI } (C)) \text{ SAU } ((j + 1 \leq n) \text{ ŞI } (D))$  atunci
       $b = 0$ 
    dacă  $b == 1$  atunci
      Tipăreşte  $i, j$ 

```

Pentru ca acest program să funcţioneze corect, condiţiile (A), (B), (C) şi (D) trebuie înlocuite cu expresii corespunzătoare. Care dintre următoarele afirmaţii sunt adevărate? Alegeţi toate variantele corecte.

Ciornă, marcaţi 1-3 răspunsuri:

① ② ③ ④ ⑤

- ① $(A) = (a[i][j] \geq a[i + 1][j])$ şi $(B) = (a[i][j] \geq a[i - 1][j])$
- ② $(A) = (a[i][j] \geq a[i - 1][j])$ şi $(B) = (a[i][j] \geq a[i + 1][j])$ ✓
- ③ $(A) = (a[i][j] < a[i - 1][j])$ şi $(B) = (a[i][j] < a[i + 1][j])$
- ④ $(C) = (a[i][j] \geq a[i][j - 1])$ şi $(D) = (a[i][j] \geq a[i][j + 1])$ ✓
- ⑤ $(C) = (a[i][j] < a[i][j - 1])$ şi $(D) = (a[i][j] < a[i][j + 1])$

Explicaţii

Pentru ca acest program să funcţioneze corect trebuie ca b să fie setat să fie 0 dacă $a[i][j]$ se învecinează direct cu o valoare mai mare sau egală. Deci b devine 0 dacă una din condiţiile următoare este satisfăcută:

- $i - 1 \geq 1$ şi $(A) = (a[i][j] \geq a[i - 1][j])$
- $i + 1 \leq n$ şi $(B) = (a[i][j] \geq a[i + 1][j])$
- $j - 1 \leq 1$ şi $(C) = (a[i][j] \geq a[i][j - 1])$
- $j + 1 \leq n$ şi $(D) = (a[i][j] \geq a[i][j + 1])$

Rezultă că variantele corecte sunt

- $(A) = (a[i][j] \geq a[i - 1][j])$ şi $(B) = (a[i][j] \geq a[i + 1][j])$
- $(C) = (a[i][j] \geq a[i][j - 1])$ şi $(D) = (a[i][j] \geq a[i][j + 1])$