

**Tema 3**  
**29 octombrie 2019**  
**Stive și cozi**

## **1. Stive**

### *Problemă obligatorie*

**Termen de predare :** Laboratorul din săptămâna 7/8 (11-15/18-22 noiembrie 2019)

(4 p) **1.** Să se implementeze cu alocare dinamică o stivă de numere întregi, cu următoarele operații:

- (a) `void push (a, stiva)` care adaugă elementul `a` în vârful stivei;
- (b) `int pop (stiva)` care scoate elementul din vârful stivei și îl întoarce ca rezultat al funcției;
- (c) `int peek(stiva)` care întoarce elementul din vârful stivei, fără a-l scoate;
- (d) `bool empty(stiva)` care verifică dacă stiva este vidă sau nu;
- (e) `int search(a, stiva)` care întoarce -1 dacă elementul `a` nu se află în stivă. Dacă `a` apare în stivă, atunci funcția întoarce distanța de la vârful stivei până la apariția cea mai apropiată de varf. Se va considera că vârful se află la distanța 0.
- (f) `void afiseaza(stiva)` care afisează stivă, pornind de la vârful ei și continuând spre bază.

### *Probleme suplimentare*

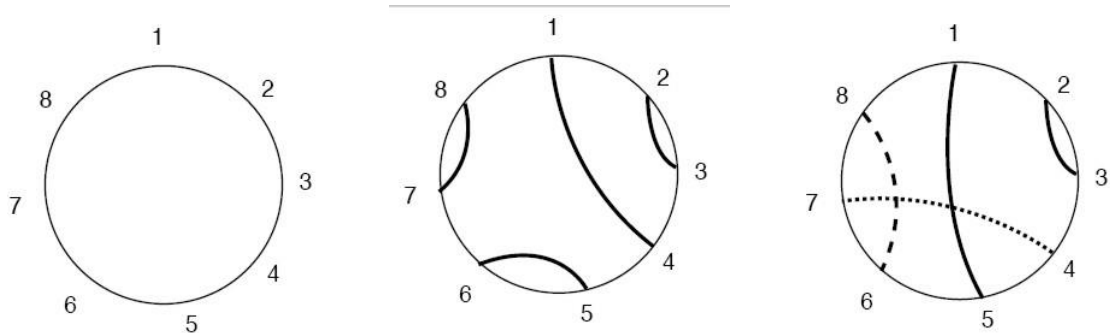
**Termen de predare :** Laboratorul din săptămâna 7/8 (11-15/18-22 noiembrie 2019)

(3p) **2.** Dat un sir  $w = w_1w_2 \dots w_n$  ( $n$  par) de caractere 'a' și 'b', să se decidă dacă în sirul  $w$  numărul de caractere 'a' este același cu numărul de caractere 'b'. Sirul de intrare se poate parcurge doar o singură dată, iar pentru a decide rezultatul se va folosi **o stivă**. Nu se permite **numărarea** aparițiilor caracterelor 'a', 'b'.

(3p) **3.** Dat un sir  $w = w_1w_2 \dots w_n$  de caractere '(' și ')', să se folosească o stivă pentru a decide dacă acest sir este corect parantezat (i.e., pentru orice subsir  $w_1 \dots w_i$ , cu  $i = \overline{1, n}$ , avem că numărul de caractere '(' este mai mare sau egal cu numărul de caractere ')'). În caz că  $w$  nu este parantezat corect, se va indica poziția primei paranteze ')' care nu are corespondent.

(3 p) 4. Consideram următoarea problema: ni se da o suprafață circulară cu un număr  $n$  de pini (tarusi) pe margini (numerotați de la 1 la  $n$ ), împreună cu o listă de perechi de pini ce trebuie conectați cu fire metalice. Problema cere să determinați în timp  $O(n)$  dacă pentru o configurație ca mai sus, pinii pereche pot fi conectați, fără ca acestea să se intersecteze. La intrare se vor citi:

- $n$  numărul de pini;
- $pereche[n]$  un vector de  $n$  componente, unde  $pereche[i] == pereche[j]$ ,  $1 \leq i < j \leq n$ , dacă pinii  $i$  și  $j$  trebuie conectați.



(a) Regiunea ce trebuie conectată (b) O configurație validă (c) O configurație invalidă

*Figura 1: Exemplu pentru problema conectării pinilor*

*Exemplul 1.* Pentru  $n = 8$  și vectorul  $pereche = (1, 2, 2, 1, 3, 3, 4, 4)$  avem configurația validă din Figura 1(b).

*Exemplul 2.* Pentru  $n = 8$  și vectorul  $pereche = (1, 2, 2, 3, 1, 4, 3, 4)$  avem configurația invalidă din Figura 1(c).

(3 p) 5. Să se scrie un algoritm pentru evaluarea unei expresii matematice în notăție postfix (forma poloneză).

## 2. Cozi

### Problemă obligatorie

**Termen de predare :** Laboratorul din săptămâna 7/8 (11-15/18-22 noiembrie 2019)

- (4 p) 6. Sa se implementeze o coada de numere intregi, cu urmatoarele operatii:
- (a) `void push(a, coada)` care adauga elementul `a` in coada;
  - (b) `int pop(coada)` care scoate primul element din coada, si il intoarce ca rezultat al functiei;
  - (c) `int peek(coada)` care intoarce primul element din coada, fara a-l scoate;
  - (d) `bool empty(coada)` care verifica daca coada este vida sau nu;
  - (e) `int search(a, coada)` care intoarce -1 daca elementul `a` nu se afla in coada. Daca `a` apare in stiva, atunci functia intoarce distanta de la primul element al cozii pana la aparitia cea mai apropiata de primul element al cozii. Se va considera ca primul element se afla la distanta 0.
  - (f) `void afiseaza(coada)` care afiseaza coada, pornind de la primul element si continuând spre ultimul.

### Probleme suplimentare

**Termen de predare :** Laboratorul din săptămâna 7/8 (11-15/18-22 noiembrie 2019)

- (3 p) 7. Spunem ca o imagine digitala binara  $M$  este o matrice de  $m \times m$  elemente (pixeli) 0 sau 1. Un element  $a$  al matricei este adiacent cu  $b$ , daca  $b$  se afla deasupra, la dreapta, dedesubtul, sau la stanga lui  $a$  in imaginea  $M$ . Spunem ca doi pixeli 1 adiacenti apartin aceleiasi componente. Problema va cere sa etichetati pixelii imaginii astfel incat doi pixeli primesc aceeasi eticheta daca si numai daca apartin aceleiasi componente.

Exemplu:

		1				
		1	1			
				1		
			1	1		
	1			1		1
1	1	1				1
1	1	1			1	1

O imagine 7 x 7

		2				
		2	2			
				3		
			3	3		
	4			3		5
4	4	4				5
4	4	4			5	5

Componentele etichetate

Figura 1: Exemplu de matrice (imagine digitală binară)

(4 p) **8.** Un depou feroviar consta dintr-o linie ferata de intrare,  $k$  linii auxiliare de depozitare, si o linie de iesire. Fiecare linie opereaza pe un sistem de coada (FIFO). In plus, vagoanele se pot deplasa doar dinspre linia de intrare spre linia de iesire (Figura 2). Sa se scrie un program care, dat un sir de vagoane pe linia de intrare (numerotate de la 1 la  $n$  si aranjate in orice ordine), descrie o strategie de a obtine pe linia de iesire sirul de vagoane  $n; n - 1; \dots; 2; 1$ , folosind liniile de depozitare.

In caz ca nu exista o astfel de strategie, se va afisa acest lucru.

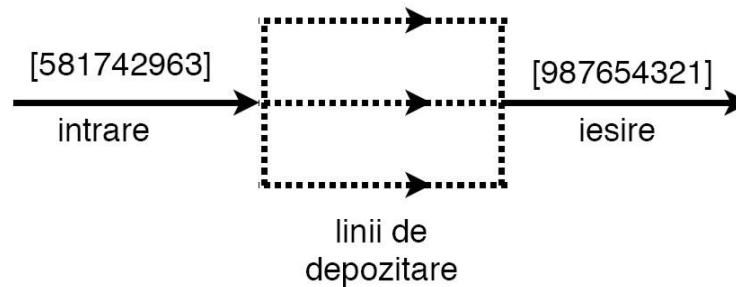


Figura 2: Exemplu de depou feroviar cu 3 linii de depozitare

### 3. Alte probleme cu liste

#### Probleme suplimentare

**Termen de predare :** Laboratorul din săptămâna 7/8 (11-15/18-22 noiembrie 2019)

(3+3 p) **9.** Dată o listă simplu înlățuită  $A$ , scrieți o funcție care inversează ordinea elementelor în listă.

Se vor aborda doua metode: prima, în care lista  $A$  nu este modificată, rezultatul afișându-se într-o nouă listă  $B$ ; a doua, în care nu se va alocă memorie suplimentară, inversarea făcându-se schimbând legăturile între nodurile listei  $A$ .

(4 p) **10.** Fie doua liste simplu înlătuite  $A$  și  $B$ , cu elementele întregi în ordine crescătoare.

Sa se creeze o noua lista simplu înlătuita  $C$  ce conține elementele din  $A$  și din  $B$  în ordine crescătoare. **Nu se va folosi memorie suplimentară.**

(4 p) **11.** Fie lista simplu înlătuita  $C$ . Sa se distribuie elementele din  $C$  în doua liste simplu înlătuite  $A$  și  $B$ , astfel:  $A$  conține elementele de pe pozițiile impare din  $C$ , iar  $B$  conține elementele din  $C$  de pe pozițiile pare. **Nu se va folosi memorie suplimentară.**

*Probleme facultative*

**Termen de predare :** Laboratorul din săptămâna 7/8 (11-15/18-22 noiembrie 2019)

Presupunem ca avem  $n$  persoane numerotate de la 1 la  $n$  dispuse pe un cerc si ca eliminam circular fiecare a doua persoana, pana cand ramane o singura persoana. Care este numarul acestei persoane?

*Exemplu:* pentru  $n = 10$ , vom elimina persoanele: 2, 4, 8, 10, 3, 7, 1, 9 (in aceasta ordine). Supravietuitorul este 5.

(5 ps) **1.** Cine este supravietuitorul pentru  $n = 2^{100} + 6$ ?

Generalizand problema de mai sus, consideram ca eliminam fiecare a  $k$ -a persoana. Cine este supravietuitorul pentru:

(5 ps) **2.**  $n = 1000$  si  $k = 7$  ? (se acorda cele 5 puncte suplimentare pentru o implementare folosind liste alocate dinamic) ?

**NOTA:** Este punctata rezolvarea unei singure probleme dintre 1 și 2.