

**Tema săptămânii 4**  
**22 octombrie 2014**

**Stive**

*Problemă obligatorie*

**Termen de predare :** Laboratorul din săptămâna 7 (14 noiembrie 2014)

(2 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 citește 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 varful 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 (14 noiembrie 2014)

(2p) **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'.

Notă : încercați o rezolvare care să implice folosirea conceptului de stivă

(2p) **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

' ) ' ). In caz ca  $w$  nu este parantezat corect, se va indica pozitia primei paranteze ' ) ' care nu are corespondent.

(2 p) **4.** Consideram urmatoarea problema: ni se da o suprafata circulara cu un numar  $n$  de pini (tarusi) pe margini (numerotati de la 1 la  $n$ ), impreuna cu o lista de perechi de pini ce trebuie conectati cu fire metalice. Problema cere sa determinati in timp  $O(n)$  daca pentru o configuratie ca mai sus, pinii pereche pot fi conectati, fara ca acestea sa se intersecteze. La intrare se vor citi:

- $n$  numarul de pini;
- $pereche[n]$  un vector de  $n$  componente, unde  $pereche[i] == pereche[j]$ ,  $1 \leq i < j \leq n$ , daca pinii  $i$  si  $j$  trebuie conectati.

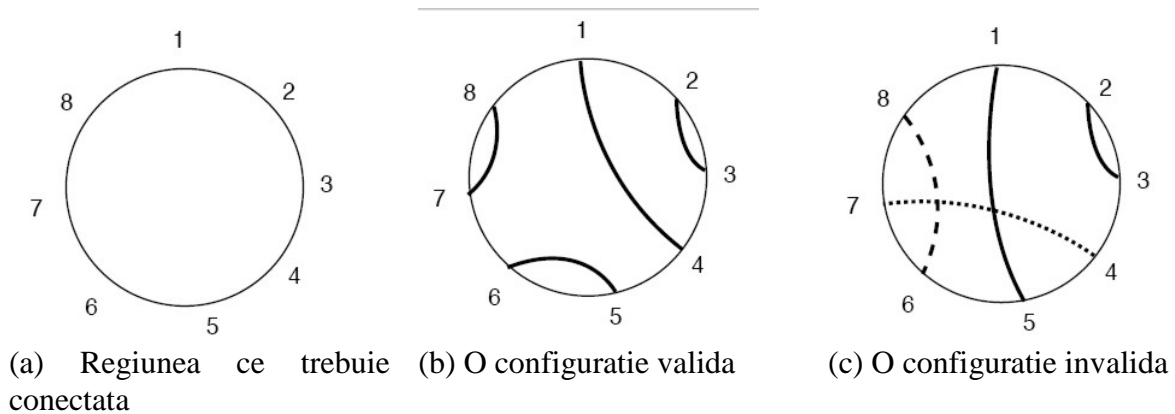


Figura 1: Exemplu pentru problema conectarii pinilor

*Exemplul 1.* Pentru  $n = 8$  si vectorul  $pereche = (1, 2, 2, 1, 3, 3, 4, 4)$  avem configuratia valida din Figura 1(b).

*Exemplul 2.* Pentru  $n = 8$  si vectorul  $pereche = (1, 2, 2, 3, 1, 4, 3, 4)$  avem configuratia invalida din Figura 1(c).

## Cozi

### Problemă obligatorie

**Termen de predare :** Laboratorul din săptămâna 7 (14 noiembrie 2014)

(2 p) 5. 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 citeste primul element din coada, fara a-l scoate;
- (d) `bool empty(coada)` care veri\_ca 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 continuand spre ultimul.

### Probleme suplimentare

**Termen de predare :** Laboratorul din săptămâna 7 (14 noiembrie 2014)

(2 p) 6. 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

(2 p) 7. 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.

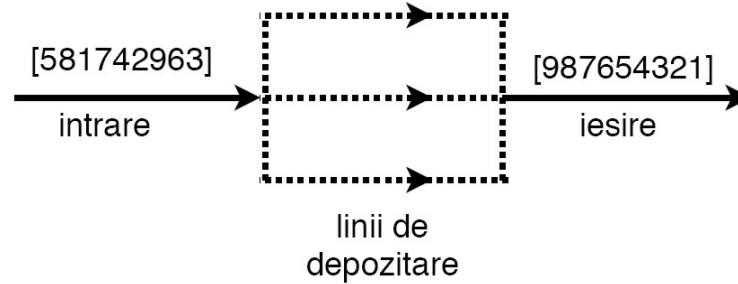


Figura 2: Exemplu de depou feroviar cu 3 linii de depozitare

*Probleme facultative*

**Termen de predare:** Laboratorul din săptămâna 5 (31 octombrie)

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.

(10 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:

(10 ps) **2.**  $n = 1000$  si  $k = 7$  ? (se acorda cele 10 puncte suplimentare pentru o implementare folosind liste alocate dinamic)

(10 ps) **3.**  $n = 104857600$  si  $k = 7$  ?

**NOTA:** Este punctata rezolvarea unei singure probleme dintre 1, 2 sau 3.