

PROBLEME - LISTE

1. Scrieți 4 proceduri cu timpul de execuție $\phi(1)$ pentru inserare elemente și ștergere de elemente la ambele capete ale unei cozi duble (complete).
2. Arătați cum se poate implementa o coadă prin 2 stive. Scrieți în Pseudocod operațiile cozii folosind doar operațiile din interfața Stivei. Analizați timpul de execuție al operațiilor cozii.
3. Arătați cum se poate implementa o stivă prin 2 cozi. Scrieți în Pseudocod operațiile cozii folosind doar operațiile din interfața Cozii. Analizați timpul de execuție al operațiilor stivei.
4. Într-o bibliotecă sunt mai multe teancuri de cărți. Bibliotecarul vrea să completeze pe un raft doar cărțile având anul de apariție mai mare decât 1970, dar în ordine alfabetică a titlurilor. Să se afișeze cărțile în ordinea în care trebuie reținute pe raft. **Indicație:** Stivă, CoadăCuPriorități.
5. **Jocul Gâsca Roșie.** 2 jucători primesc inițial $\frac{n}{2}$ cărți de joc (fiecare carte poate avea culoarea roșie sau neagră). Jucătorii pun alternativ câte o carte pe masă (din vârful teancului lor de cărți), până se pune o carte roșie (caz în care teancul de pe masă va fi luat de către jucătorul care nu a pus cartea roșie și adăugat sub teancul său de cărți). Pierde jucătorul care nu mai are cărți. Simulați jocul. **Indicație:** Stivă, Coadă.
6. **Problema lui Joseph.** n persoane se află în cerc: persoanele se elimină din m în m , începând cu persoana cu numărul k . Se cere să se afișeze ordinea în care vor fi eliminate persoanele din cerc. **Indicație:** Listă (circulară).
7. Fie un labirint (rețea dreptunghiulară) cu celule ocupate (X) și libere (*). Fie R un robot în acest labirint. Robotul se poate deplasa în 4 direcții: N, S, E, V.

*	*	X	*	*
X	*	*	*	X
X	*	R	*	*
X	*	X	X	*
*	*	*	*	*

- a). Testați dacă R poate ieși din labirint (poate ajunge la margine).
- b). Determinați un drum pentru ieșire (dacă există).
- c). Determinați un drum de lungime minimă pentru ieșire (dacă există).

Indicație

Fie T mulțimea pozițiilor în care robotul poate ajunge pornind de la poziția inițială. Notăm cu S mulțimea pozițiilor în care robotul a ajuns până la un moment dat și din care s-ar putea deplasa. Un algoritm pentru determinarea mulțimilor T și S ar putea fi:

```

 $T \leftarrow \{\text{poziția inițială}\}$ 
 $S \leftarrow \{\text{poziția inițială}\}$ 
Cât-timp  $S \neq \emptyset$  execută
    Fie  $p$  un element din  $S$ 
     $S \leftarrow S \setminus \{p\}$ 
    Pentru fiecare poziție  $q$  alăturată poziției  $p$ ,  $q \neq 'X'$  și  $q \notin T$  execută
         $S \leftarrow S \cup \{q\}$ 
         $T \leftarrow T \cup \{q\}$ 
    SfPentru
SfCatTimp

```

Observații

- Pentru a răspunde la punctul a), algoritmul s-ar putea termina dacă poziția q care satisface condițiile este pe frontiera labirintului.
- Mulțimea T poate fi memorată printr-o matrice asociată labirintului (ex: 0 pentru pozițiile neatinse încă, respectiv 1 pentru pozițiile în care robotul a ajuns).
- Structura S poate fi o Stivă sau o Coadă.
- Pentru a răspunde la punctul b), ne putem gândi la un algoritm care pornind de la o poziție de pe frontieră (în cazul în care răspunsul la punctul a) a fost pozitiv) merge din aproape în aproape pe pozițiile marcate cu 1 (atinse) spre poziția inițială.