Stive și cozi

1 Stive

O **stivă** este o listă în care alocarea se realizează în mod dinamic pe principiul **LIFO** - *last in, first out*. Astfel ultimele elemente adăugate în listă vor fi primele eliminate.

1.1 Inserarea unui element

Adăugarea unui element nou în cadrul stivei se realizează în vârf (acesta va deveni **Top**).

```
void Push (int Val) {
    Node *p = (Node*) malloc (sizeof(Node));
    if (p == NULL)
        // Overflow
    else {
        p -> Data = Val;
        p -> Next = Top;
        Top = p;
    }
}
```

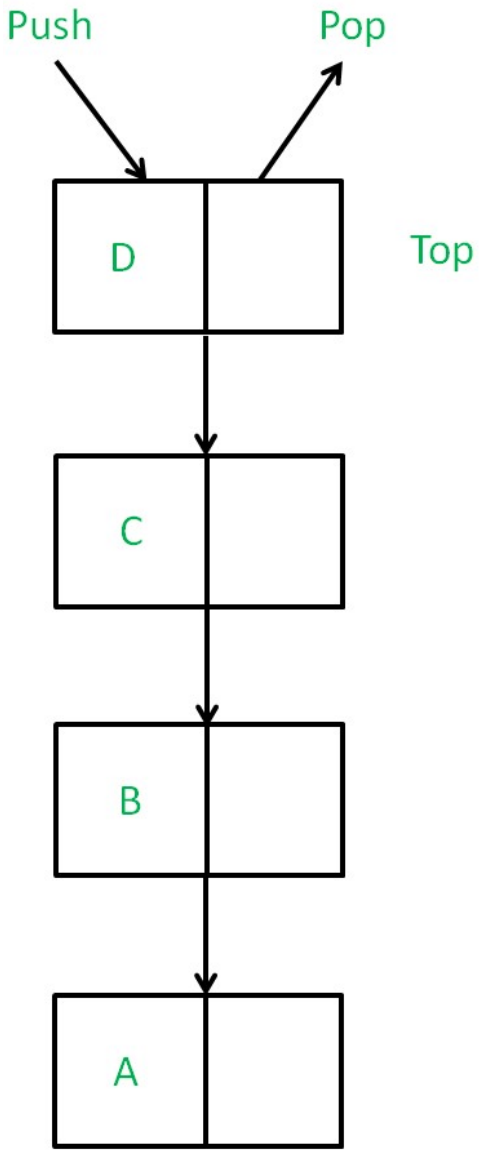

Figura 1: Reprezentarea unei stive

1.2 Ștergerea unui element

Ștergerea unui element se face din vârful stivei. Nodul următor va deveni **Top**.

```
void Pop (){
    Node * Temp;
        if (Top == NULL)
                // Underflow
    else {
        Temp = Top;
        X = Temp -> Data;
        Top = Top -> Next;
        free (Temp);
    }
}
```

2 Cozi

O **coadă** este o listă în care alocarea se realizează în mod dinamic pe principiul **FIFO** - *first in, first out*. Astfel primele elemente adăugate în listă vor fi primele eliminate.

2.1 Inserarea unui element

Adăugarea unui element nou în cadrul cozii se realizează la coadă (acesta va deveni **Rear**).

```
void Insert (int Val) {
   Node *p = (Node *) malloc(sizeof(Node));
   if (p == NULL)
    // Overflow
   else {
    p -> Data = Val; p -> Next = NULL;
    if (Rear == NULL)   Front = p;
    else    Rear -> Next = p;
    Rear = p;
  }
}
```

2.2 Ștergerea unui element

Ștergerea unui element se face de la începutul cozii. Nodul următor va deveni **Front**.

```
void Delete (){
    if (Front == NULL)
            // Underflow
    else {
        X = Front -> Data;
        Node *Temp = Front;
        Front = Front -> Next;
        free(Temp);
        if (Front == NULL)  Rear = NULL;
    }
}
```

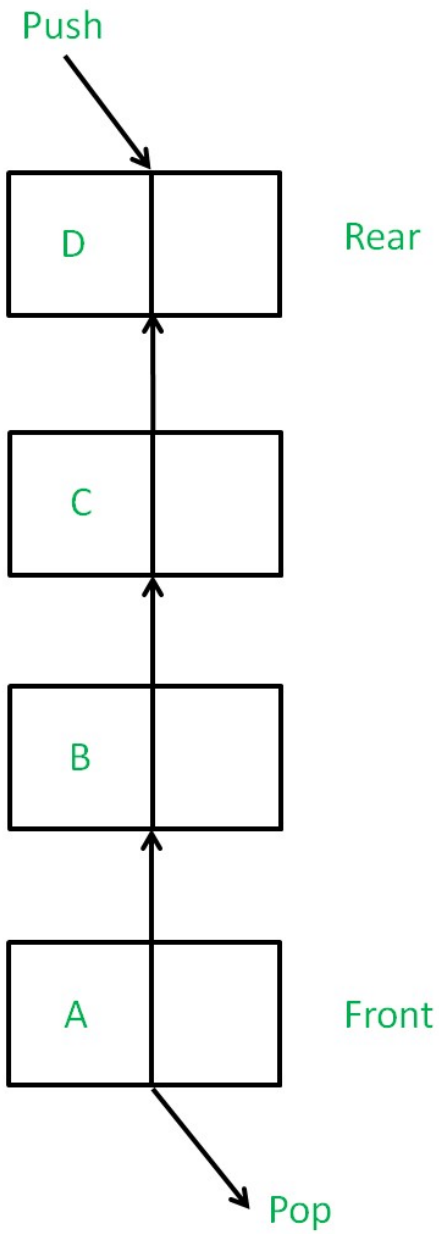

Figura 2: Reprezentarea unei cozi

3 Aplicații

Exercițiul 1 - Coadă cu priorități. (2.5p) O **coadă cu priorități** este o coadă în care elementele au, pe lângă cheie şi o prioritate. Vom presupune că cea mai înaltă prioritate este 1, urmată de 2 etc. Ordinea liniară este dată de regulile:

- elementele cu aceeaşi prioritate sunt extrase (şi procesate) în ordinea intrării;
- toate elementele cu prioritate i se află înaintea celor cu prioritate $i+1$ (şi deci vor fi extrase înaintea lor).

Implementați operațiile de inserare și de ștergere pentru acest tip de structură.

Exercițiul 2. (2.5p) Se consideră un string s reprezentând o expresie validă formată din numere și '+', '-', '(', ')', ' '. Implementați un calculator pentru evaluarea expresiei și returnați rezultatul. Folosiți una dintre structurile din laborator. **Atenție!** '+' nu este folosit ca operație unară (i.e., `+1` și `+(2 + 3)` sunt invalide). '-' poate fi folosit ca operație unară (i.e., $-$`1` și $-$`(2 + 3)` sunt valide).

Exemple:

s = "1 + 1" -> 2
s = " 2-1 + 2 " -> 3
s = "(1+(4+5+2)-3)+(6+8)" -> 23

Exercițiul 3. (2.5p) Considerăm următoarea problemă: ni se dă o suprafață circulară cu un număr n de pini (țăruși) 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 perechile să se intersecteze. Folosiți una dintre structurile învățate. La intrare se vor citi:

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

Exemple:
Pentru n = 8 și vectorul pereche = (1, 2, 2, 1, 3, 3, 4, 4) avem configurația validă din Figura 3(b).
Pentru n = 8 și vectorul pereche = (1, 2, 2, 3, 1, 4, 3, 4) avem configurația invalidă din Figura 3(c).

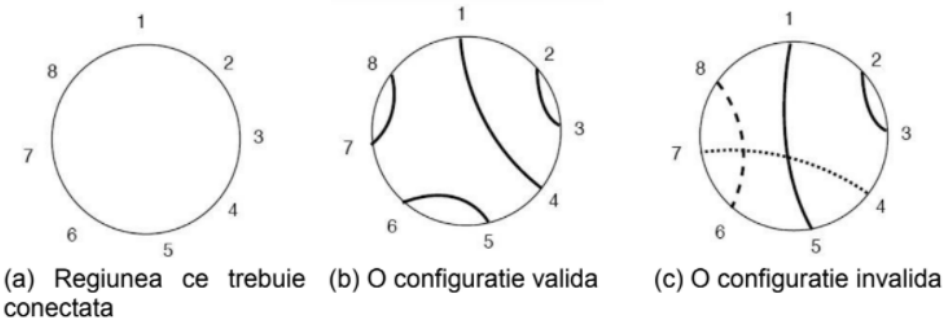

Figura 3: Exemplu pentru problema conectării pinilor

Exercițiul 4. (2.5p) Spunem că o imagine digitală binară M este o matrice de m x m elemente (pixeli) 0 sau 1. Un element a al matricei este adiacent cu b dacă b se află deasupra, la dreapta, dedesubtul, sau la stânga lui a în imaginea M. Spunem că doi pixeli 1 adiacenți aparțin aceleiași componente. Problema vă cere să etichetați pixelii imaginii astfel încât doi pixeli să primească aceeași etichetă dacă și numai dacă aparțin aceleiași componente. Folosiți una dintre structurile învățate.

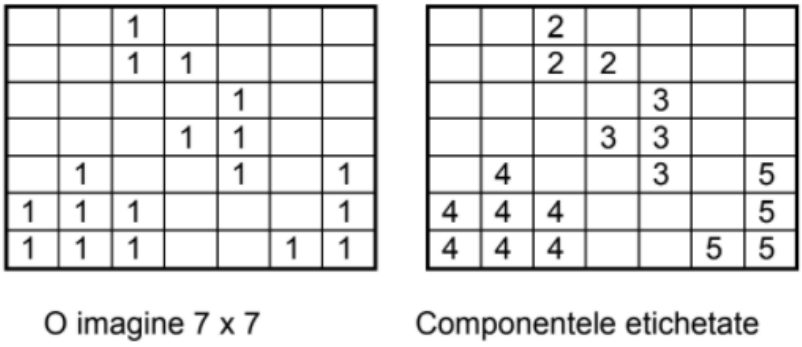

Figura 4: Exemplu de matrice (imagine digitală binară)

Exercițiul 5. (2.5p) Un depou feroviar constă dintr-o linie ferată de intrare, 3 linii auxiliare de depozitare și o linie de ieșire. Fiecare linie operează pe un sistem de coadă (FIFO). În plus, vagoanele se pot deplasa doar dinspre linia de intrare spre linia se ieșire (Figura 5). Să se scrie un program care, dat un sir de vagoane pe linia de intrare (numerotate de la 1 la n și aranjate în orice ordine), descrie o strategie de a obține pe linia de ieșire șirul de vagoane n; n - 1;... ; 2; 1, folosind liniile de depozitare. În caz că nu există o astfel de strategie, se va afișa acest lucru..

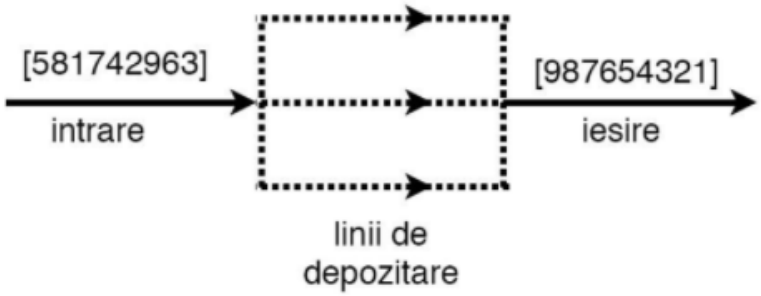

Figura 5: Exemplu depou feroviar cu 3 linii de depozitare

Exercițiul 6. (2.5p) Se dau n numere naturale reprezentând o hartă cu denivelări de lățime 1. Să se calculeze câtă apă poate fi stocată în cadrul formei obținute. Folosiți una dintre structurile învățate.
Exemple:
Pentru harta [0,1,0,2,1,0,1,3,2,1,2,1], se vor obține 6 unități de apă.
Pentru harta [4,2,0,3,2,5], se vor obține 9 unități de apă.

Figura 6: Exemplu pentru [0,1,0,2,1,0,1,3,2,1,2,1]

4 Observații

1. Punctajul necesar pentru Laboratorul 4 se poate obține prin realizarea exercițiilor 1 - 4, exercițiile 5 și 6 fiind bonus.

2. Prezentarea problemei (problemelor) se poate realiza în timpul laboratoarelor L și $L+1$ unde L este considerat laboratorul curent (în cazul de față laboratoarele 4 și 5).

3. Prezentarea se poate desfășura atât fizic în timpul laboratorului, cât și online printr-o programare stabilită anterior cu laborantul (pentru cazuri speciale: simptome COVID, probleme personale etc.)

4. Transmiterea laboratorului se realizează pe adresa de e-mail ruxandra.balucea@unibuc.ro pana în ziua laboratorului. Denumirea exercițiilor va fi de tipul x_grupa_Nume_Prenume (exemplu: 2_141_Pop_Ion). Toate aceste fișiere vor fi transmise într-o arhivă **grupa_Nume_Prenume.zip**.