STRUCTURI DE DATE

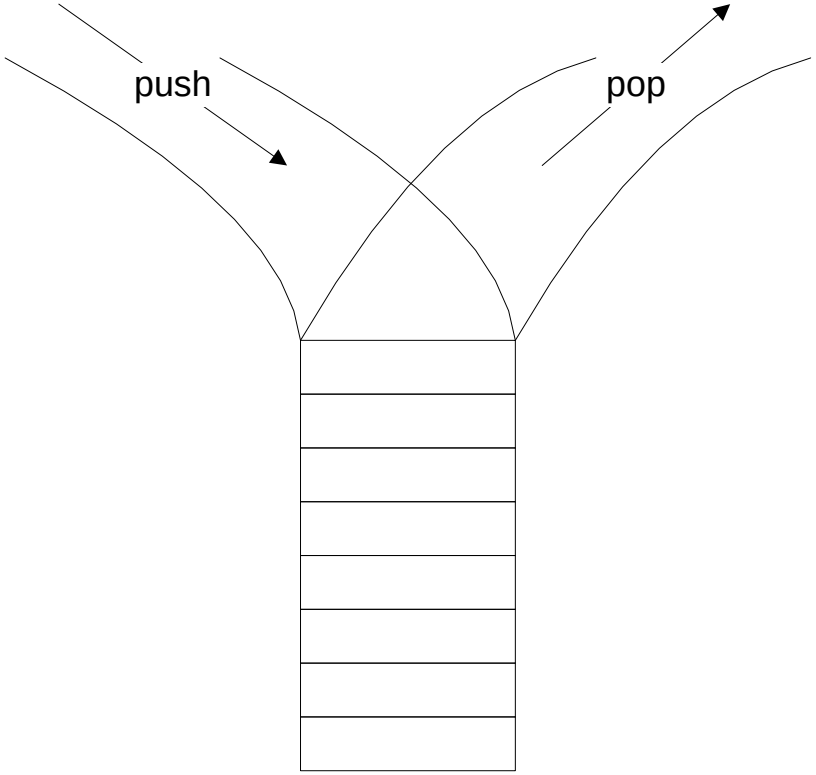
Stiva
Coada

STIVA

Structura de tip stiva:

- Structura de date **logica**: implementarea este făcută utilizând alte structuri de date.
- Structura de date **omogena**: toate elementele sunt de acelaşi tip.
- Două **operaţii de bază**: adăugarea şi extragerea unui element.
- **Disciplina de acces**: LIFO-Last In First Out - toate inserările (push) şi extragerile (pop) sunt făcute la unul din capetele structurii de implementare (inceput lista simpla), denumit vârful stivei.

STIVA



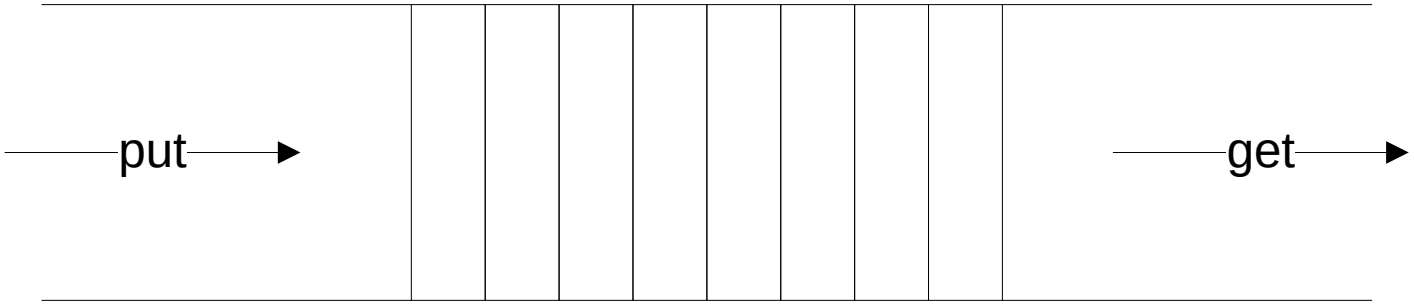
Mecanismul de stiva

COADA

Structura de tip coada:

- Structura de date **logica:** implementarea este făcută utilizând alte structuri de date;
- Structura de date **omogena:** toate elementele sunt de acelaşi tip;
- Două **operaţii de bază:** adăugarea şi extragerea unui element;
- **Disciplina de acces:** FIFO-First In First Out - toate inserările (put) se fac la un capat (sfarsit lista simpla) şi extragerile (get) sunt făcute la celalalt capat (inceput lista simpla).

COADA



Mecanismul de coadă

STIVE SI COZI - APLICATII

Conversia unei valori zecimale in format binar

1. Preluare valoare zecimala.
2. Cat timp valoarea este strict pozitiva:
    2.1 determinare rest impartire la 2.
    2.2 prelucrare rest – utilizare structura stiva.
    2.3 determinare cat impartire la 2.

STIVE SI COZI - APLICATII

QuickSort

- Algoritm de tip divide-et-impera.
- Reducerea fiecarui set de valori de sortat in 2 sub-seturi;
- Aplicarea operatiei de reducere pe fiecare din cele 2 sub-seturi;
- Utilizarea a 2 structuri de tip stiva (**Inf**, **Sup**) pentru stocarea pozitiilor celor 2 sub-seturi.
- Stivele sunt initializate: **Inf = {1}**, **Sup = {10}**.

STIVE SI COZI - APLICATII

QuickSort – exemplu:

Faza de reducere:

extragere varf de stiva Inf (index 1) si varf de stiva Sup (index 10)

identificare pozitie finala a unei valori din set (ex. valoare pozitie 1).

Incepand cu pozitia 10, se cauta de la dreapta la stanga prima valoare mai mica decat 13.

13	85	25	9	64	99	93	49	72	20

STIVE SI COZI - APLICATII

QuickSort – exemplu:

Valoare identificata este 9.

Se interschimba **13 cu** 9.

Incepand cu pozitia lui 9, se cauta de la stanga spre dreapta prima valoare mai mare decat **13, pana la pozitia lui 13**.

STIVE SI COZI - APLICATII

QuickSort – exemplu:

Valoare identificata este **85**.

Se interschimba **85** cu **13**.

Incepand cu pozitia lui **85**, se cauta de la dreapta la stanga prima valoare mai mica decat **13**, **pana la pozitia lui 13**.

STIVE SI COZI - APLICATII

QuickSort – exemplu:

Nu exista nici o valoare, deci pozitia finala a lui **13** este determinata (**2**).

Stivele sunt populate pentru cele 2 subintervale.

Subintervalul stang contine 1 element, deci este sortat. Nu se retine nimic pe stive.

Subintervalul drept contine cel putin 2 elemente. Stivele sunt: **Inf = {3 }** si **Sup = {10 }**.

STIVE SI COZI - APLICATII

QuickSort – exemplu:

Faza de reducere:

extragere varf de stiva Inf (index 3) si varf de stiva Sup (index 10)

identificare pozitie finala a primei valori din set 25.

Incepand cu pozitia 10, se cauta de la dreapta la stanga prima valoare mai mica decat 25.

STIVE SI COZI - APLICATII

QuickSort – exemplu:

Valoare identificata: **20**.

Se interschimba **25** cu **20**.

Incepand cu pozitia lui **20**, se cauta de la stanga la dreapta prima valoare mai mare decat **25**.

STIVE SI COZI - APLICATII

QuickSort – exemplu:

Valoare identificata: **85**.

Se interschimba **25** cu **85**.

Incepand cu pozitia lui **85**, se cauta de la dreapta la stanga prima valoare mai mica decat **25**.

STIVE SI COZI - APLICATII

QuickSort – exemplu:

Valoare identificata: **-**

Este determinata pozitia lui **25** in setul final (ordonat), respectiv pozitia **4**.

Stivele sunt populate pentru cele 2 subintervale.

Subintervalul stang contine 1 element, deci este sortat. Nu se retine nimic pe stive.

Subintervalul drept contine cel putin 2 elemente. Stivele sunt: **Inf = {5}** si **Sup = {10 }**.

STIVE SI COZI - APLICATII

QuickSort – exemplu:

Faza de reducere:

- Extragere varf de stiva **Inf** (index **5**) si varf de stiva **Sup** (index **10**).
- Identificare pozitie finala a primei valori din set **64**.

Incepand cu pozitia **10**, se cauta de la dreapta la stanga prima valoare mai mica decat **64**.

STIVE SI COZI - APLICATII

QuickSort – exemplu:

- Faza de reducere se aplica recursiv.
- Oprire algoritm: stivele **Inf** si **Sup** sunt empty.
- Stivele sunt utilizate pentru a retine limitele de intervale obtinute din modul recursiv de aplicare a algoritmului.

STIVE SI COZI - APLICATII

Evaluarea expresiilor matematice ce utilizeaza ca structură de date principală stiva:

- Rearanjarea expresiei într-o anumită formă astfel încât ordinea operaţiilor să fie clară şi evaluarea să necesite o singură parcurgere a expresiei.

- Forma poloneză: matematicianul de origine poloneză Jan Lukasiewicz.

STIVE SI COZI - APLICATII

Evaluarea expresiilor matematice (continuare):

- Forma poloneză: scrierea operatorilor înaintea operanzilor.

- Forma poloneză inversă: operatorii sunt scrişi în urma operanzilor.

STIVE SI COZI - APLICATII

Forma poloneză inversă (scriere postfixata): avantaje faţă de scrierea prefixată (forma poloneza) sau infixată (expresia matematica):

- Ordinea în care se efectuează operaţiile este clară.
- Parantezele nu mai sunt necesare.
- Evaluările sunt uşor de efectuat cu ajutorul calculatorului.

STIVE SI COZI - APLICATII

Algoritm de transformare din expresie matematică în scriere postfixată:

- Edsger Dijkstra (algoritmul macazului – Dijkstra Shunting Algorithm).
- Utilizare stivă în care sunt păstraţi operatorii şi din care sunt eliminaţi şi transferaţi în scrierea postfixată.
- Fiecare operator are atribuită o ierarhie după cum este prezentat în tabelul urmator.

STIVE SI COZI - APLICATII

Operator	Ierarhie
([{	1
)] }	2
+ -	3
* /	4

Ierarhia operatorilor

STIVE SI COZI - APLICATII

Expresia matematică (scriere infixată)	Expresia în forma poloneză (scriere prefixată)	Expresia în forma poloneză inversă (scriere postfixată)
4 + 5	+ 4 5	4 5 +
4 + 5 * 5	+ 4 * 5 5	4 5 5 * +
4 * 2 + 3	+ * 4 2 3	4 2 * 3 +
4 + 2 + 3	+ + 4 2 3	4 2 + 3 +
4 * (2 + 3)	* 4 + 2 3	4 2 3 + *

Forme ale scrierii unei expresii matematice

STIVE SI COZI - APLICATII

Algoritmul este:

- Se iniţializează stiva şi scrierea postfixată.
- Atât timp cât nu s-a ajuns la sfârşitul expresiei matematice:
  - se citeşte următorul element din expresie.
  - dacă este valoare se adaugă în scrierea postfixată.
  - dacă este **(** se introduce în stivă.
  - dacă este **)** se transferă elemente din stivă în scrierea postfixată până la **(**.

STIVE SI COZI - APLICATII

- altfel:

a. atât timp cât ierarhia operatorului din vârful stivei este mai mare ierarhia operatorului curent, se trece elementul din vârful stivei în scrierea postfixată;

b. se introduce operatorul curent în stivă.

- Se trec toţi operatorii rămaşi pe stivă în scrierea postfixată.

STIVE SI COZI - APLICATII

Algoritmul de evaluare:

- Se iniţializează stiva.
- Atât timp cât nu s-a ajuns la sfârşitul scrierii postfixate:
  - se citeşte următorul element.
  - dacă este valoare se depune pe stivă.
  - altfel (este operator):
    a. se extrage din stivă elementul y;
    b. se extrage din stivă elementul x;
    c. se efectuează operaţia x operator y;
    d. se depune rezultatul pe stivă;
- Ultima valoare care se află pe stivă este rezultatul expresiei.