

Lucru de laborator № 2

Tema lucrării laboratoare:

Elaborarea unui mecanism de planificare a activității sistemului de operare

Partea teoretică:

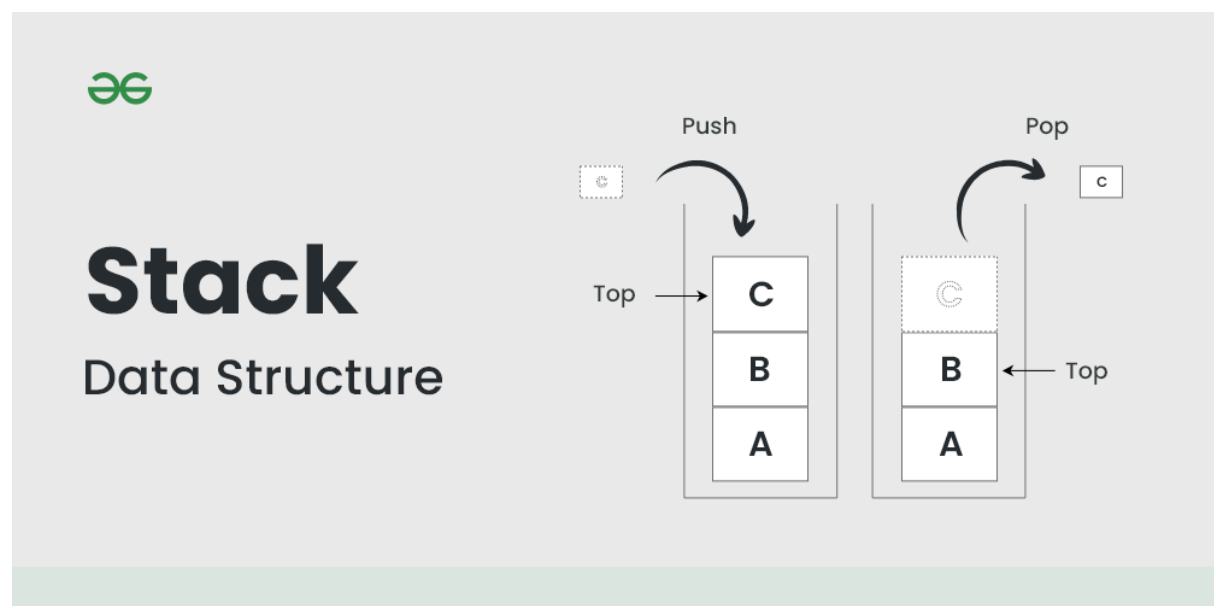
Ce este o stivă?

O stivă este o structură de date liniară în care inserarea unui element nou și ștergerea unui element existent au loc la același capăt, reprezentat sub forma unui vârf de stivă.

Pentru a implementa o stivă, trebuie să menținem un pointer la vertexul stivei care reprezintă ultimul element care urmează să fie inserat, deoarece putem accesa numai elementele din partea de sus a stivei.

LIFO (Ultimul intrat primul ieșit)

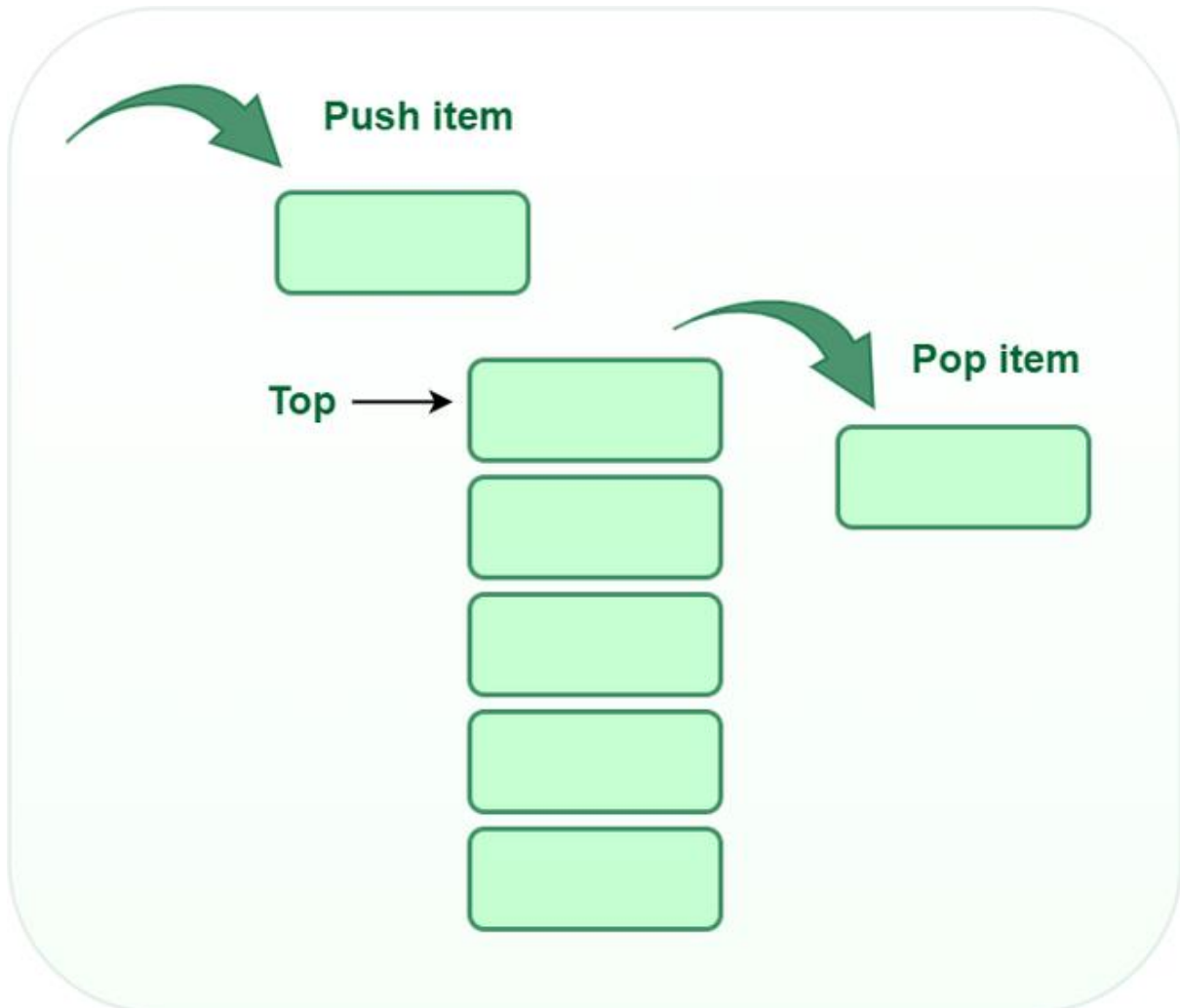
Această strategie prevede că elementul care a fost introdus ultimul va ieși primul. Ca exemplu, putem lua un teanc de farfurii așezate unele peste altele. Farfuria pe care am introdus-o ultima se află deasupra și, pe măsură ce scoatem farfuria de deasupra, putem spune că farfuria care a fost introdusă ultima iese prima.



Operații de bază pe stivă

Pentru a efectua manipulări în stivă, avem la dispoziție anumite operații.

- push() - inserarea unui element în stivă
- pop() - scoaterea unui element din stivă
- top() - returnează elementul superior al stivei.
- isEmpty() - returnează true dacă stiva este goală, iar în caz contrar false.
- size() - returnează dimensiunea stivei.



- **Push**

Adaugă un element în stivă. În cazul în care stiva este depășită, se spune că stiva se află într-o stare de depășire.

Algoritm pentru push:

```
begin
  if stack is full
    return
  endif
else
  increment top
  stack[top] assign value
end else
end procedure
```

- **Pop**

Îndepărtarea unui element din stivă. Elementele sunt scoase în ordinea opusă celei în care au fost introduse. În cazul în care stiva este goală, se consideră că aceasta a fost depășită.

Algoritm pentru pop:

```
begin
  if stack is empty
    return
  endif
else
  store value of stack[top]
  decrement top
  return value
end else
end procedure
```

- **Top**

Returnează elementul superior al stivei.

Algoritm pentru Top:

```
begin
  return stack[top]
end procedure
```

- **isEmpty**

Se returnează adevărat dacă stiva este goală, altfel fals.

Algoritm pentru isEmpty:

```
begin
  if top < 1
    return true
  else
    return false
  end if
end procedure
```

O înțelegere practică a stivei:

Există multe exemple de stivă în viața reală. Luați în considerare un exemplu simplu de stivă de farfurii așezate unele peste altele într-o sufragerie. Farfuria care se află în partea de sus este îndepărtată prima, adică farfuria care a fost așezată pe locul cel mai de jos rămâne în stivă cel mai mult timp. Astfel, putem spune că în stivă se respectă ordinea LIFO/FILO.

Tipuri de stive

- - Stivă de dimensiuni fixe - după cum sugerează și numele, o stivă de dimensiuni fixe are o dimensiune fixă și nu poate fi mărită sau redusă în mod dinamic. În cazul în care stiva este plină și se încearcă adăugarea unui element, apare o eroare de depășire. În cazul în care stiva este goală și se încearcă să se elimine un element din ea, apare o eroare de depășire.
- - Stiva dinamică - o stivă dinamică poate fi mărită sau micșorată în mod dinamic. Atunci când stiva este umplută, dimensiunea sa este mărită automat pentru a găzdui noul element, iar atunci când stiva este golită, dimensiunea sa este redusă automat. Acest tip de stivă este implementat folosind o listă legată, deoarece permite redimensionarea ușoară a stivei.

Realizarea stivei:

O stivă poate fi implementată folosind o matrice sau o listă legată. În cazul unei implementări bazate pe un tablou, o operațiune de împingere se efectuează prin creșterea indicelui elementului superior și prin stocarea noului element la acel indice. Operațiunea pop se efectuează prin descreșterea indicelui elementului superior și returnarea valorii stocate la acel indice. Într-o implementare bazată pe o listă legată, operațiunea de împingere se realizează prin crearea unui nou nod cu un nou element și prin stabilirea următorului pointer al nodului superior curent la acel nod. Operațiunea pop se realizează prin stabilirea următorului pointer al nodului superior curent la nodul următor și returnarea valorii nodului superior curent.

Stivele sunt utilizate pe scară largă în informatică pentru o varietate de sarcini, inclusiv evaluarea expresiilor, apelarea funcțiilor și gestionarea memoriei. La evaluarea expresiilor, stiva poate fi utilizată pentru a stoca operanzi și operatori pe măsură ce aceștia sunt evaluați. La apelarea funcțiilor, stiva poate fi utilizată pentru a ține evidența ordinii în care sunt apelate

funcțiile și pentru a returna controlul către funcția corectă atunci când aceasta se întoarce. În gestionarea memoriei, stiva poate fi utilizată pentru a stoca valorile contorului de program și ale registrelor într-un program de calculator, permițând programului să revină la o stare anterioară atunci când se întoarce o funcție.

Source: <https://www.geeksforgeeks.org/introduction-to-stack-data-structure-and-algorithm-tutorials/>

Sarcina principală:

Implementați un algoritm operat pe stivă care să aducă informații, să proceseze informațiile fie înainte de a le pune pe stivă, fie pentru a le prelua de pe stivă, și să scoată din stivă această sau acea informație.

Opțiuni de realizare a misiunii:

- **Opțiune ușoară:**

Prefix - O expresie este numită prefixată dacă operatorul apare înaintea operanzilor din expresie.

Exemplu: $*+AB-CD$ (Infix: $(A+B) * (C-D)$)

Postfix - O expresie se numește postfix dacă operatorul apare în expresie după operanzi.

Exemplu: $AB+CD-*$ (Infix: $(A+B * (C-D))$)

După ce ați definit o expresie prefixă, convertiți-o într-o expresie postfixă.

- **Opțiune medie:**

Se dă o secvență formată din primele n numere naturale în ordine aleatorie.

Folosii stiva pentru a aranja elementele acestei secvențe în ordine crescătoare.

- **O opțiune dificilă:**

Având în vedere un tablou cu n elemente și q interogări, pentru fiecare interogare care are un index de element, găsiți următorul element mai mare și afișați valoarea acestuia. În cazul în care nu există un astfel de element mai mare la dreapta acestuia, se afișează -1 .

- **Opțiune ușoară:**

Postfix - o expresie se numește postfix dacă operatorul se află în expresie după operanzi.

Exemplu: $AB+CD-*$ (Infix: $(A+B * (C-D))$)

Prefix - o expresie se numește prefixată dacă operatorul apare în expresie înaintea operanzilor.

Exemplu: $*+AB-CD$ (Infix: $(A+B) * (C-D)$)

După ce ați definit o expresie postfixă, convertiți-o într-o expresie prefixă.

Opțiune medie:

Turnul din Hanoi este un puzzle matematic. Este format din trei piloni și mai multe discuri de diferite dimensiuni care pot fi înșirate pe orice pilon. Puzzle-ul începe cu discurile care sunt stivuite ordonat în ordine crescătoare a mărimii pe un stâlp, cu cel mai mic disc în vârf, formând astfel o formă de con. Sarcina puzzle-ului este de a muta toate discurile de pe un stâlp (să zicem, "inițial") pe altul (să zicem, "final") cu ajutorul unui al treilea stâlp (să zicem, auxiliar).

În puzzle se aplică următoarele două reguli:

- Nu puteți așeza un disc mai mare deasupra unui disc mai mic.
- Nu se poate muta decât un singur disc la un moment dat.

O opțiune dificilă:

La o petrecere formată din N persoane, o singură persoană este cunoscută de toată lumea. Această persoană poate fi prezentă la petrecere, iar dacă este așa, nu cunoaște pe nimeni de la petrecere. Putem pune doar întrebări de genul "A îl cunoaște pe B?". Găsiți necunoscutul (celebritatea) în numărul minim de întrebări.

- **Opțiune ușoară:**

Infix: O expresie se numește expresie Infix dacă operatorul se află între operanzii din expresie.

Exemplu: $(A+B) * (C-D)$

Prefix: O expresie se numește expresie prefix dacă operatorul se află înaintea operanzilor din expresie.

Exemplu: $+AB-CD$ (Infix: $(A+B) (C-D)$).

Având în vedere o expresie prefixă, convertiți-o într-o expresie infixă.

Opțiune medie:

Sortați stiva cu ajutorul stivei temporare în ordine crescătoare sau descrescătoare.

O opțiune dificilă:

Găsirea parantezelor duplicate în stivă. Un set de paranteze este considerat duplicat dacă aceeași subexpresie este înconjurată de mai multe paranteze. Exemplu:
 $((a+b)+((c+d)))$

- **Opțiune ușoară:**

Efectuați o inversare a stivei prin transferarea dintr-o stivă în alta

Opțiune medie:

Efectuați o inversare a stivei, elementele rămânând în cele din urmă pe stiva inițială.

O opțiune dificilă:

Efectuați o inversare a stivei folosind bubble sort.

- **Opțiune ușoară:**

Determinați toate numerele pare sau impare din stivă.

Opțiune medie:

Identificați toate numerele din stivă care corespund unei anumite formule matematice.

O opțiune dificilă:

Determinați toate numerele perfecte din stivă. (Verificați printr-o formulă, fără o matrice pregătită)

- **Opțiune ușoară:**

Identificați toate numerele perfecte de pe stivă.

Opțiune medie:

Îndepărtați elementul din mijloc al stivei.

O opțiune dificilă:

Găsirea următorului mai mic al următorului mai mare dintr-o matrice

După finalizarea activității, scrieți un raport, care trebuie să includă - numele, prenumele, grupul, tema de lucru, varianta de implementare a sarcinii, o scurtă descriere a implementării sarcinii, un link către codul sursă pe GitHub. Împingeți codul sursă în ramura dvs. în depozitul corespunzător - <https://github.com/FCIM-SO/Practice-Work-RO>. Salvați raportul în format PDF și trimiteți-l la ELSE - <https://else.fcim.utm.md/mod/assign/view.php?id=43454>.