

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
CATARINENSE
Campus Videira



Pilha

Professor: Manassés Ribeiro

manasses.ribeiro@ifc.edu.br



Agenda

- Conceitos
- Operações
- Aspectos de Implementação usando Vetores
- Aspectos de Implementação usando Lista Encadeada



Conceitos



Conceito

Definição 1:

- É um conjunto de dados ordenados no qual **novos itens só podem ser inseridos e removidos a partir de uma única extremidade chamada **topo** da pilha.**





Conceito

Definição 2:

- É um **objeto dinâmico**, **constantemente mutável** que compreende a inserção e remoção de itens por uma única **extremidade** designada como o **topo** da pilha.





Conceito

- Neste tipo de TAD, novos elementos **só podem ser colocados no topo da pilha**, bem como, elementos já existentes só podem ser eliminados a partir do **topo da pilha**;
- Por esta razão, ocasionalmente uma pilha é chamada lista **Last-in, First-out (LIFO)** ou seja, **o último a entrar é o primeiro a sair**.



Operações



Operações

- As duas operações básicas de uma TAD pilha são:
empilhar e **desempilhar**:
 - Quando um item é **inserido** em uma pilha ele é **empilhado**: função **push()**;
 - Quando um item é **removido** de uma pilha ele é **desempilhado**: função **pop()**.



Portanto:

- Em função de uma pilha s e de um item i , executar a função ***push(s, i)*** incluirá o item i no topo da pilha s .
- De modo semelhante, a operação ***pop(s)*** removerá o elemento superior e o retornará como valor da função.



Operações

- Não existe um limite máximo de itens que podem ser mantidos numa pilha, contudo, uma pilha sem nenhum item é considerada uma pilha vazia:
 - O limite em uma pilha é determinado pelo limite físico da estrutura que está sendo utilizada na implementação (vetor ou ADM);
 - Caso extrapole o limite físico da estrutura ocorre erro de ***overflow***.



Operações

- Com isto, a operação de empilhamento (*push*) pode ser utilizada em qualquer situação da pilha, inclusive pilha vazia.
- Por outro lado, a operação que desempilha (*pop*) **não pode ser aplicada a uma pilha vazia** uma vez que esta pilha não tem elementos a serem desempilhados (erro de *underflow*).



Operações

- Além das básicas, outras funções úteis podem ser definidas, tais como:
 - ***empty(s)***: verifica se uma pilha está vazia ou não. Caso verdadeiro retorna TRUE, caso falso, FALSE.
 - ***stacktop(s)***: retorna a consulta ao elemento que está no **topo** da pilha **s** (mas não remove, apenas consulta).



Operações

- Da mesma maneira que a função ***pop()***, a função ***stacktop()*** não pode ser aplicada a uma pilha vazia (erro de ***underflow***).
- O resultado de uma tentativa inválida de **desempilhar**, ou acessar um item de uma pilha vazia, (erro de ***underflow***), pode ser evitado assegurando que ***empty()*** seja falso antes de tentar as operações ***pop()*** ou ***stacktop()***.



Aspectos de Implementação usando Vetores



Aspectos de Implementação

- Da mesma forma que na implementação de filas, as pilhas podem ser implementada utilizando **vetores** ou **alocação dinâmica de memória**;
- No primeiro caso a implementação fica **limitada ao tamanho do vetor** e pode ser útil em situações onde se conhece ou pode ser pré-determinado o tamanho da pilha.



Aspectos de Implementação

- No caso de se implementar uma pilha utilizando vetores, esta pode ser declarada como estrutura contendo dois objetos:
 - **um vetor** para armazenar os **elementos** da pilha; e
 - **um inteiro** para indicar a posição atual do **topo**.



Aspectos de Implementação

- Pode-se definir uma pilha **s** de inteiros com:

```
#define STACKSIZE 100
```

Constante que
define o
tamanho
máximo da fila



Aspectos de Implementação

- Pode-se definir uma pilha **s** de inteiros com:

```
#define STACKSIZE 100  
typedef struct {  
    int top;  
    int items[STACKSIZE];  
} Stack;
```

TAD da
Pilha



Aspectos de Implementação

- Pode-se definir uma pilha **s** de inteiros com:

```
#define STACKSIZE 100  
typedef struct {  
    int top;  
    int items[STACKSIZE];  
} Stack;  
  
Stack s;  
s.top = -1;
```

TAD da
Pilha

Inicialização
da Pilha



Aspectos de Implementação

- Inicialmente:
 - **s.top** é definido com -1.
- A pilha está vazia sempre que **s.top < 0**;
- O número de elementos na pilha, a qualquer momento, é igual ao valor de **s.top + 1**.



Aspectos de Implementação

- Ignorando a possibilidade de ***underflow*** e ***overflow***, a operação ***push(s, i)*** poderia ser implementada pelas instruções:

$s.items[++s.top] = i;$

- onde ***i*** é o elemento a ser inserido na pilha.

- e a operação ***pop(s)*** poderia ser implementada por:

$x = s.items[s.top--];$

- onde ***x*** é o elemento removido e retornado.

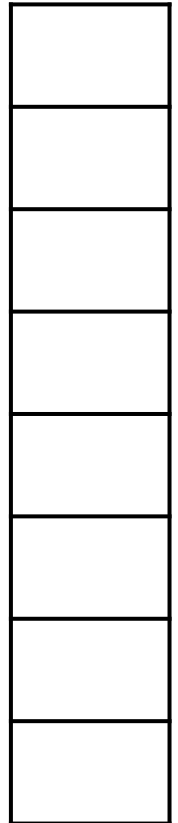


Exemplo de operações

Operação:

Pilha vazia

s =



$s.top = -1$

$size(s.top + 1) = -1 + 1 = 0$

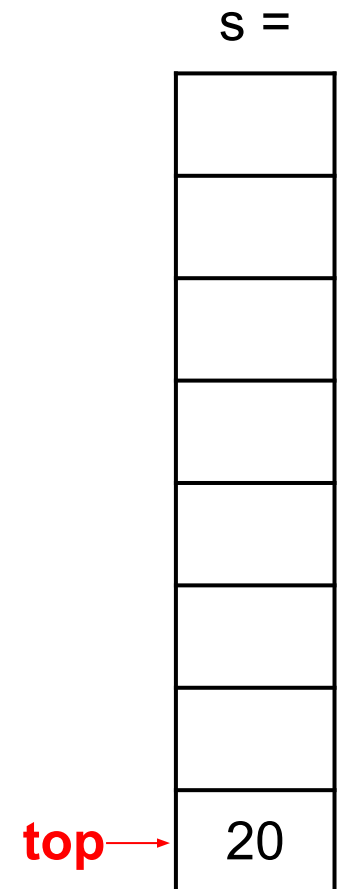


Exemplo de operações

Operação: ***push*** (**s**, 20)

$s.top = 0$

$size(s.top + 1) = 0 + 1 = 1$



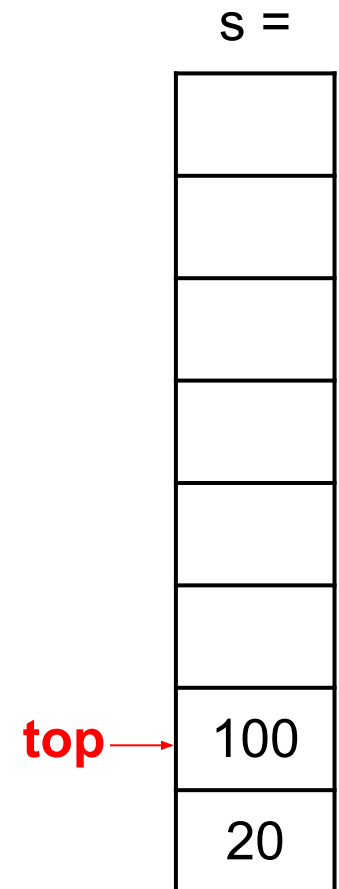


Exemplo de operações

Operação: *push* (s, 100)

s.top = 1

size (s.top + 1) = 1 + 1 = 2



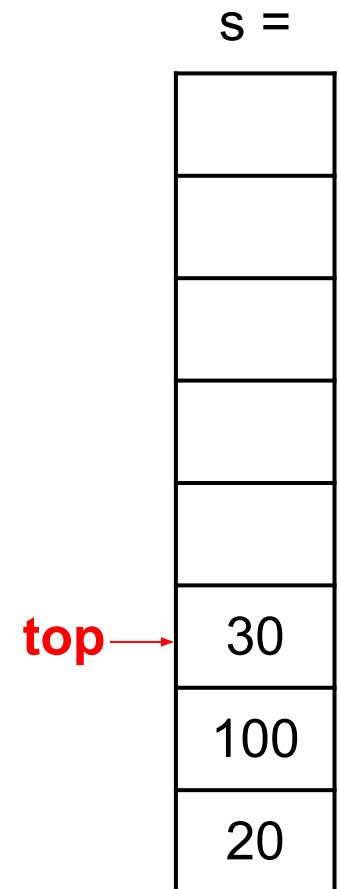


Exemplo de operações

Operação: *push* (s, 30)

s.top = 2

size (s.top + 1) = **2 + 1 = 3**



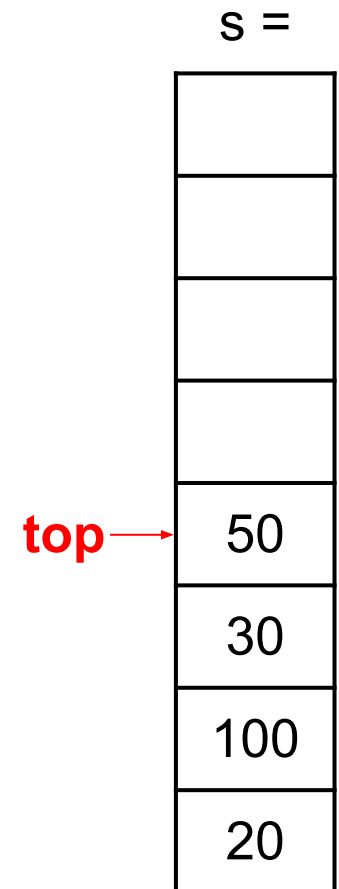


Exemplo de operações

Operação: *push* (s, 50)

s.top = 3

size (s.top + 1) = **3 + 1 = 4**



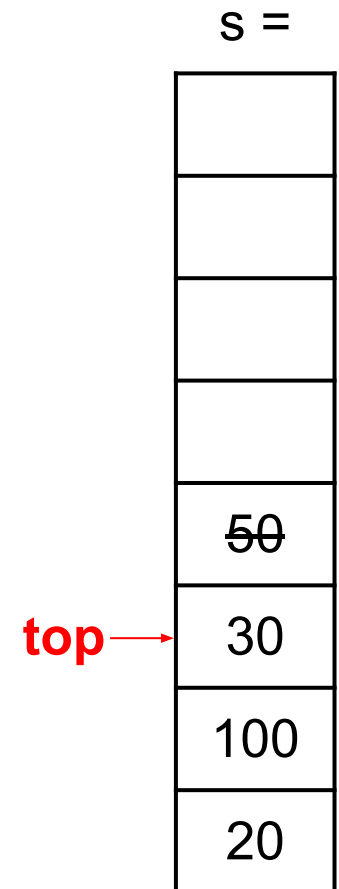


Exemplo de operações

Operação: ***pop*** (s) \Rightarrow 50

s.top = 2

size (s.top + 1) = **2 + 1 = 3**



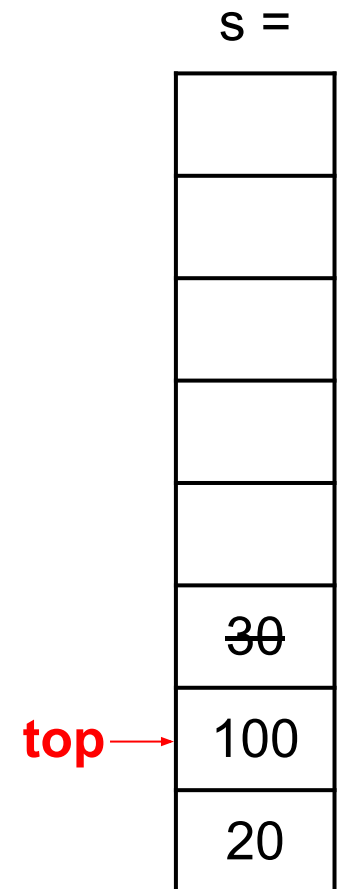


Exemplo de operações

Operação: ***pop*** (s) \Rightarrow 30

s.top = 1

size (s.top + 1) = 1 + 1 = 2





Aspectos de Implementação usando Lista Encadeada



Aspectos de Implementação

- A outra opção de implementação da pilha é utilizando **lista encadeada dupla**;
- Nesta opção a entrada na pilha (função ***push()***) será sempre pelo ***tail*** da lista que também será o **topo da pilha**;



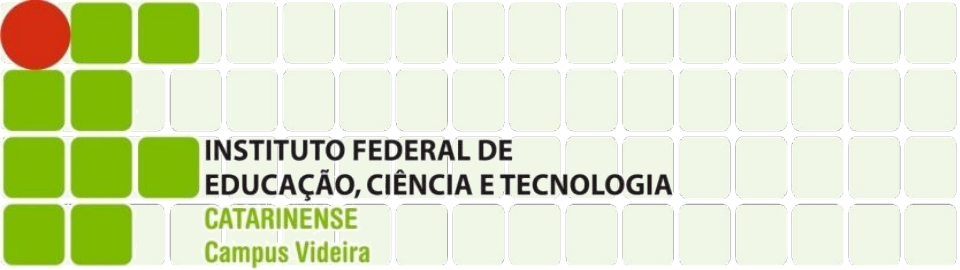
Aspectos de Implementação

- Da mesma forma, a função ***pop()***, que remove elementos da lista, acontecerá pelo ponteiro ***tail*** da lista;
- A função ***empty()*** poderá ser obtida por meio da variável *size* que indica o tamanho da lista.



Resumo

- Conceitos
- Operações
- Aspectos de Implementação usando Vetores
- Aspectos de Implementação usando Lista Encadeada



Pilha

Professor: Manassés Ribeiro

manasses.ribeiro@ifc.edu.br