

GOVERNO DO ESTADO DO RIO DE JANEIRO
UERJ – UNIVERSIDADE DO ESTADO DO RIO DE JANEIRO - ZO
CTC – CENTRO DE TECNOLOGIAS E CIÊNCIAS
FCEE – FACULDADE DE CIÊNCIAS EXATAS E ENGENHARIAS
DEPCOMP – DEPARTAMENTO DE COMPUTAÇÃO
CURSOS DE COMPUTAÇÃO

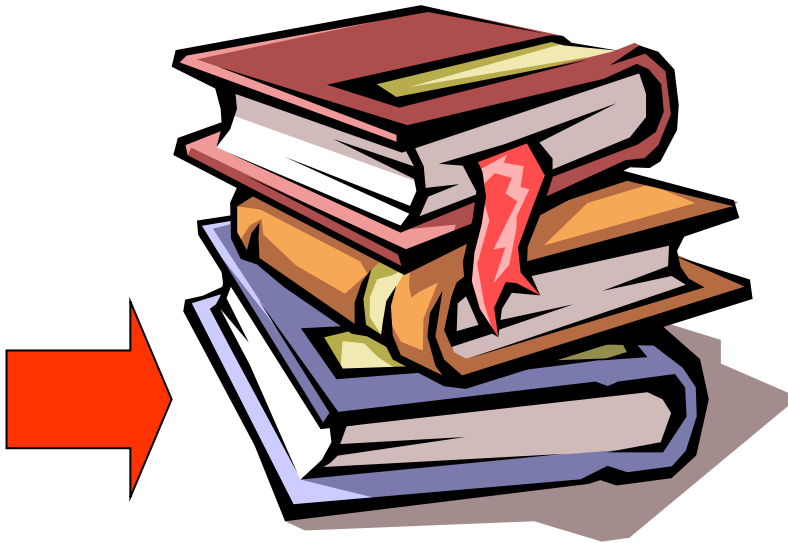
Estruturas de Dados
Parte 5 – Pilhas Estáticas.
Prof. Raul Queirós

Estruturas de Dados

Pilhas Estáticas.

Estruturas de Dados

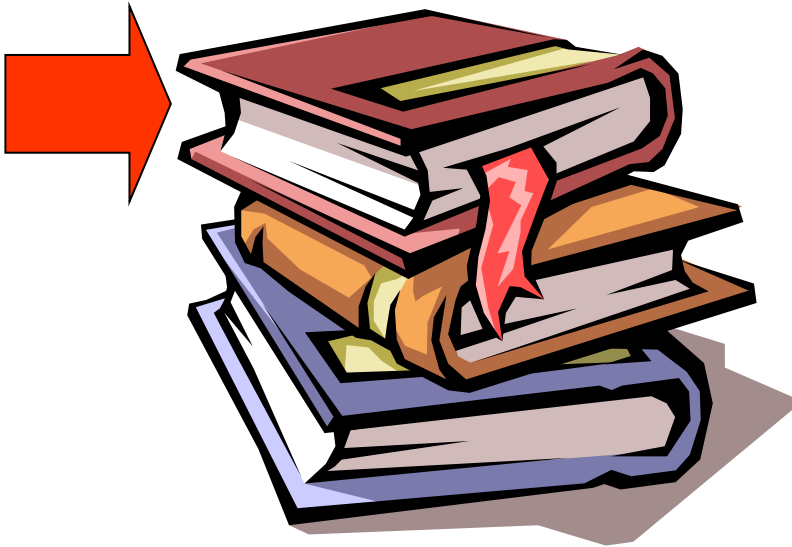
- Pilha é uma estrutura de dados em que as operações de inserção e remoção são realizadas em uma mesma extremidade, denominada TOPO.



Qual é o primeiro livro da pilha ?

Estruturas de Dados

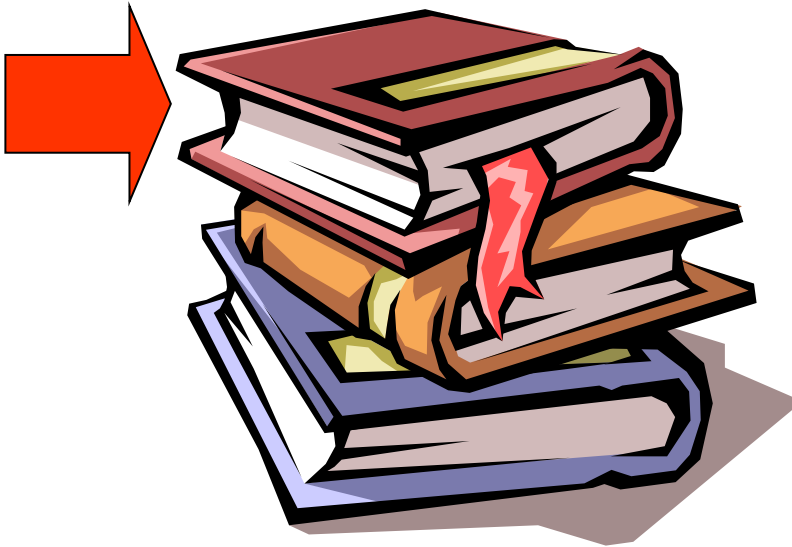
- Pilha é uma estrutura de dados em que as operações de inserção e remoção são realizadas em uma mesma extremidade, denominada TOPO.



Qual é o último livro da pilha ?

Estruturas de Dados

- Pilha é uma estrutura de dados em que as operações de inserção e remoção são realizadas em uma mesma extremidade, denominada TOPO.



Qual livro está no TOPO da pilha ?

Estruturas de Dados

Pilhas (L. I. F. O.)

A pilha também é conhecida como estrutura LIFO - **L**ast **I**n **F**irst **O**ut

Último a entrar, primeiro a sair.

Dentre os elementos que ainda permanecem no conjunto, o primeiro elemento a ser retirado é o último que foi inserido.

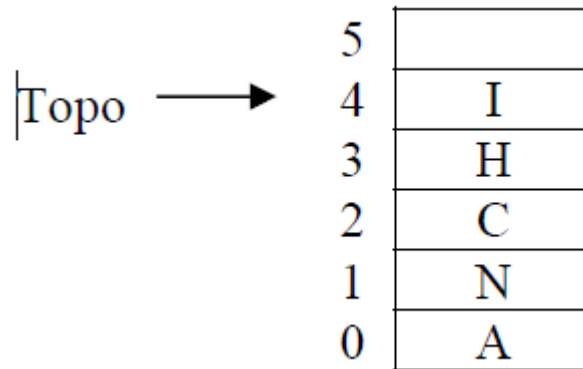
Estruturas de Dados

Pilha (Stack)

É uma lista linear, em que todas as inserções e remoções de elementos só podem ser feitas numa extremidade chamada Topo.

`inserir('A');` `inserir('N');` `inserir('C');` `inserir('H');` `inserir('I');`

`push('A');` `push('N');` `push('C');` `push('H');` `push('I');`



`var = retirar();` `// I`

`var = pop();` `// I`

Estruturas de Dados

Implementação Estática de uma Pilha.

A maneira mais simples de se representar uma pilha é através de um vetor de n elementos.

Digamos que este vetor tem tamanho n e chama-se \mathbf{P} e seus elementos tem índices $0, 1, 2, \dots, n-1$.

O número máximo de elemento da pilha será n , o elemento do fundo será $P[0]$, o elemento seguinte $P[1]$ e assim por diante. Devido a este fato, esta representação é chamada seqüencial.

Associada a pilha tem-se uma variável chamada Topo , de tal forma que o elemento no topo da pilha será $P[\text{Topo}]$.

Pode-se convencionar que $\text{Topo} = -1$ indica uma pilha vazia.

Estruturas de Dados

Operações associadas com uma pilha:

- Inicializar uma pilha P vazia;
- Inserir (push) na pilha P;
- Excluir (pop) da Pilha P;
- Verificar se a pilha P está vazia;
- Verificar se a pilha está cheia.

Exemplo de uso de Pilhas;

- Processamento das chamadas de sub-rotinas e seus retornos;
- Simulação de sistemas;
- Sequencias bem-formadas;
- Retorno de páginas de navegadores;
- Controles de telas de aplicativos em dispositivos móveis;
- Representação de precedência sem o uso de parênteses em expressões aritméticas;
- Análise de expressões Pós fixas (Calculadoras HP 12C).

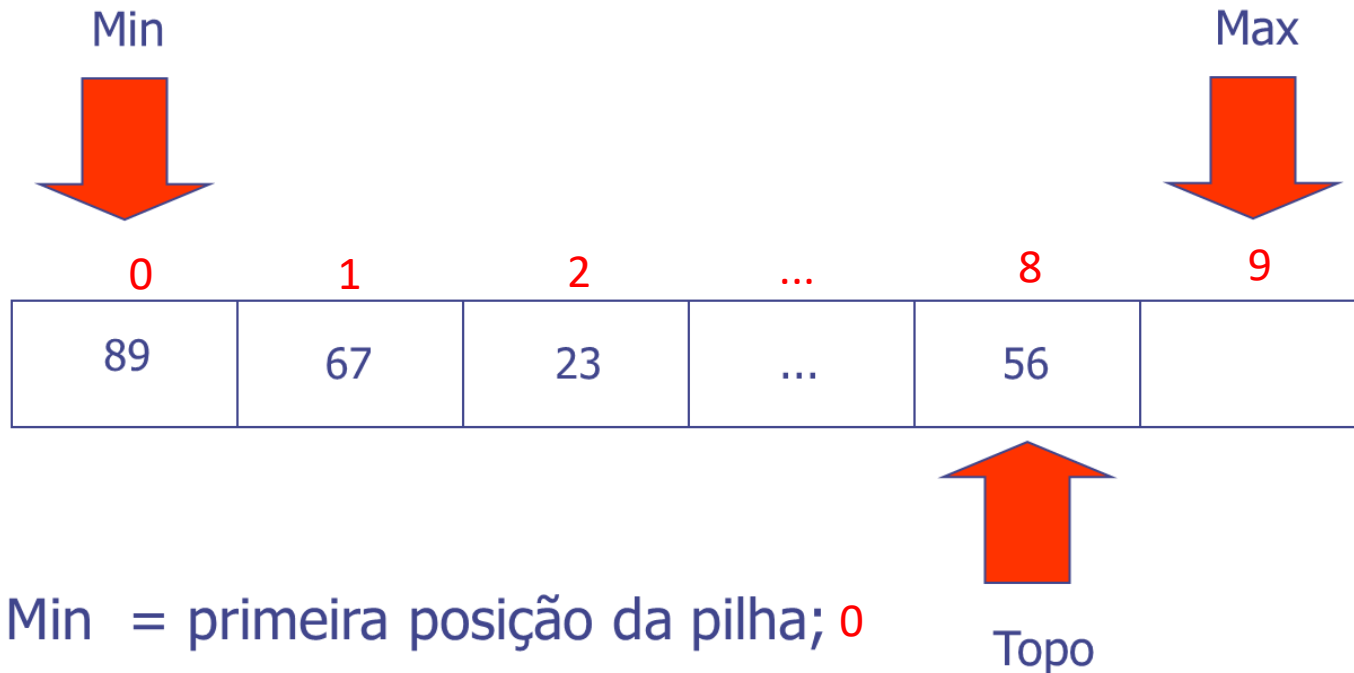
Estruturas de Dados

- A inserção e a retirada só podem ser realizadas no TOPO da pilha;

Somente DUAS operações:

- EMPILHAR (PUSH)
(inserir um novo elemento no topo da pilha).
- DESEMPILHAR (POP)
(remover o elemento do topo da pilha).

Pilhas Seqüenciais



Min = primeira posição da pilha; 0

Max = última posição da pilha; 9

$(\text{Max} - \text{Min} + 1) = \text{número de elementos da pilha. } 10$

Estruturas de Dados

Algoritmos para Pilha utilizando vetores

- Convenções:

Topo = -1 indica pilha vazia

Topo = TAM_VET - 1 indica pilha cheia

- Inicialização

Topo = -1

Estruturas de Dados

- INSERÇÃO (push)

Se $\text{Topo} = \text{TAM_VET} - 1$ então

“PILHA CHEIA - INSERÇÃO IMPOSSÍVEL”

senão

$\text{Topo} \leftarrow \text{Topo} + 1$

$P[\text{Topo}] \leftarrow \text{Elemento}$

Fimse

Estruturas de Dados

- REMOÇÃO (pop)

Se Topo = -1 então

“PILHA VAZIA - REMOÇÃO IMPOSSÍVEL”

senão

Elemento $\leftarrow P[\text{Topo}]$

Topo $\leftarrow \text{Topo} - 1$

Fimse

Estruturas de Dados

Exemplo prático:

Implementação de pilhas estáticas com vetores em linguagem C.

Pilha de reais (float)

```
#include <stdio.h>
#include <stdlib.h>
#include "locale.h"

#define TAM_VET 10
#define VERD 1
#define FALSO 0
#define ERRO -1
```

Estruturas de Dados

```
void iniciaPilha (int *Topo) {  
    *Topo = -1;  
}
```

```
int pilhaVazia(int *Topo) {  
    if (*Topo == -1) return(VERD);  
    return(FALSO);  
}
```

```
int pilhaCheia(int *Topo) {  
    if (*Topo == TAM_VET - 1) return(VERD);  
    return(FALSO);  
}
```


Estruturas de Dados

```
int push(float P[], int *Topo, float valor) {  
    if (pilhaCheia(Topo)) {  
        printf ( "Erro - Ins. Pilha cheia");  
        return(ERRO);  
    }  
    else {  
        (*Topo)++;  
        P[*Topo] = valor;  
    }  
    return(VERD);  
}
```

Estruturas de Dados

```
int pop(float P[], int *Topo, float *valor) {  
    if (pilhaVazia(Topo)) {  
        printf ( "Erro - Pilha vazia");  
        return(ERRO);  
    }  
    else {  
        *valor = P[*Topo];  
        (*Topo)--;  
    }  
    return(VERD);  
}
```

Estruturas de Dados

```
void exibirVetor(float P[], int *Topo){
    int i;
    printf ( "\nExibir elementos preenchidos do Vetor\n");
    for(i=0;i<=*Topo;i++){
        printf ( "\nElemento [%d]: %.2f ", i, P[i]);
    }
    printf ( "\n\n");
}
```

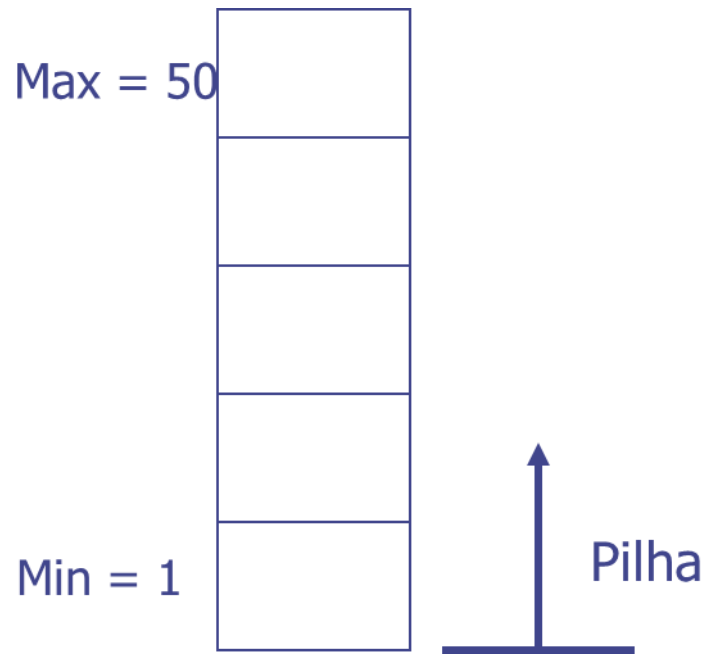
Estruturas de Dados

```
int main(){
    setlocale(LC_ALL, "Portuguese");
    float P[TAM_VET];
    int Topo;
    int ok, op=-1;
    float dado;
    iniciaPilha(&Topo);
    while ( op != 0 ) {
        printf ( "\n\nMenu\n1-Inserir\n2-Excluir\n3-Exibir Vetor\n0-Fim\nOpcao: ");
        scanf ("%d", &op);
        if ( op == 1 ) {
            printf ( "\nDigite um número: ");
            scanf ("%f", &dado);
            push(P,&Topo,dado);
        }
        else if ( op == 2 ) {
            ok = pop(P,&Topo, &dado);
            if ( ok != ERRO)
                printf ( "\nElemento removido: %.2f ", dado);
        }
        else if ( op == 3 ) {
            exibirVetor(P,&Topo);
        }
    }
    return 0;
}
```

Estruturas de Dados

- Exercício Prático 1:

Suponha a seguinte estrutura de pilha sequencial (inteiros):



Escreva um programa em linguagem C, com as funções de empilhar e desempilhar os elementos da Pilha, além das funções de inicialização e verificação de Pilha Cheia e Vazia.

Estruturas de Dados

- Exercício Prático 2:

De acordo com o exercício anterior, crie uma Pilha para armazenamento de uma pequena agenda telefônica de 10 elementos, capaz de armazenar o nome, o telephone e a idade das pessoas.

Estruturas de Dados

Obrigado pela atenção!

Fim da parte 5.