

TAD - Pilha

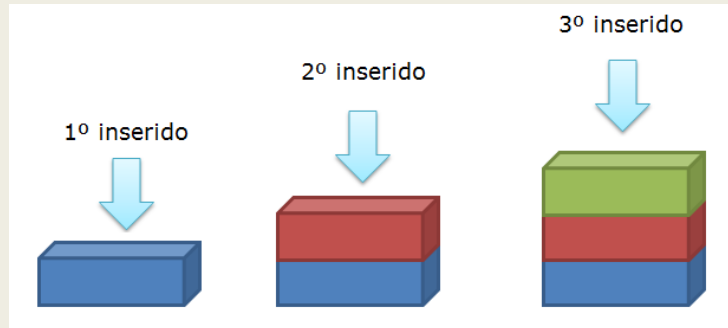
Andressa Andrade, Guilherme Bernal, Renata Antunes e Rodrigo Fernandes

O que é um TAD? E uma Pilha?

- TAD

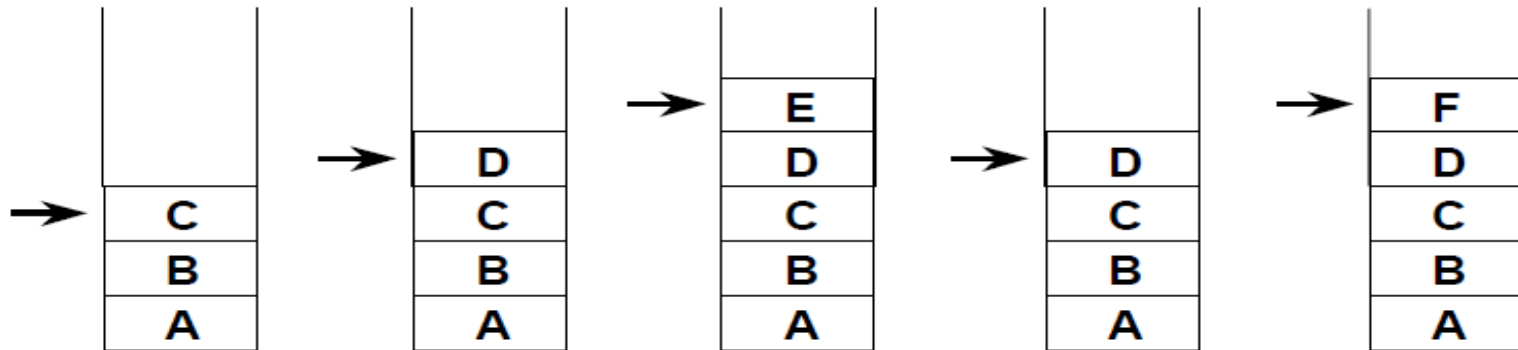
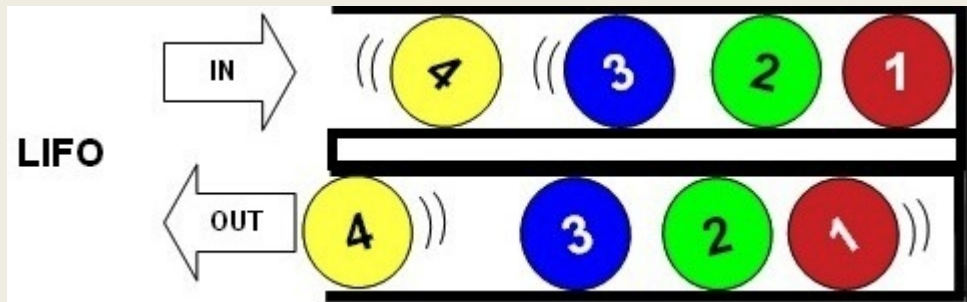


- Pilha



Propriedades da pilha

- Estrutura simples
- Lifo



Curiosidades sobre a pilha

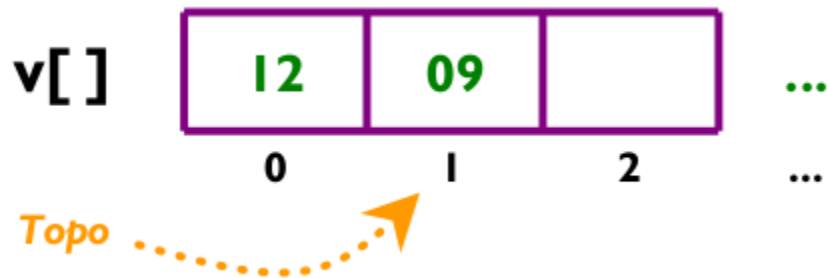
- Ideal para estruturas aninhadas
- Recursividade
- Usadas para:
 - Mecanismos de desfazer/refazer dos editores de texto
 - Navegação entre páginas Web

Alocação

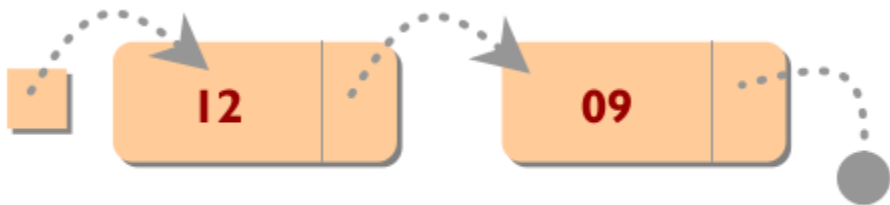
- Estática
- Sequencial
- Dinâmica

Implementação da pilha

- Vetores
- Estruturas encadeadas



TPilhaEnc p*



Operações da Pilha

- `stack_alloc`
- `stack_free`
- `stack_push`
- `stack_pop`
- `stack_sort`
- `stack_first`
- `stack_top`
- `stack_nth`
- `stack_size`

Complexidade

Acesso ao Topo:

Tempo constante – $O(1)$

- Inserir
- Remover
- Espaço
- Desvantagem

Stack.h

```
#pragma once
```

```
typedef struct stack_tstack;
```

```
// create empty
```

```
// O(1)
```

```
stack* stack_alloc(void);
```

```
// free whole stack
```

```
// O(size)
```

```
void stack_free(stack* st);
```

```
// add element, read top, remove top
```

```
// O(1)
```

```
void stack_push(stack* st, void* el);
```

```
void* stack_top(stack* st);
```

```
void* stack_pop(stack* st);
```

```
//O(n log n)
```

```
void stack_sort(stack* st);
```

```
// read first element
```

```
// O(1)
```

```
void *stack_first(stack* st);
```

```
// get nth element
```

```
// O(n)
```

```
void* stack_nth(stack* st, unsigned i);
```

```
// get size
```

```
// O(1)
```

```
unsigned stack_size(stack* st);
```

Stack_alloc

entrada: nenhuma

processo: aloca a pilha e a coloca na condição de vazia

saida: endereço da pilha

Stack_free

entrada: endereço da pilha

processo: libera toda area alocada para a pilha

saida: nenhuma

Stack_push

entrada: endereço da pilha e o elemento

processo: insere elemento no topo da pilha e
atualiza pilha

saida: 1 se sucesso , 0 se fracasso

Stack_pop

entrada: endereço da pilha

processo: remove elemento do topo da pilha e atualiza pilha

saida: endereço do elemento no topo da pilha ou NULL se pilha é vazia

Stack_sort

entrada: endereço da pilha

processo: ordena a pilha

saida: pilha ordenada

Stack_top

entrada: endereço da pilha

processo: consulta o topo da pilha

saida: endereço do elemento no topo da pilha
ou NULL se pilha é vazia

Stack_first

entrada: endereço da pilha

processo: consulta a base da pilha

saida: endereço do elemento na base da pilha
ou NULL se pilha é vazia

Stackelement_nth

entrada: endereço da pilha, índice

processo: buscar um elemento específico na pilha

saida: elemento desejado

Stack_size

entrada: endereço da pilha

processo: verificar tamanho da pilha

saida: tamanho da pilha ou NULL se pilha está vazia