

PILHAS

DCE792 - AEDs II (Prática)

Atualizado em: 6 de setembro de 2023

Iago Carvalho

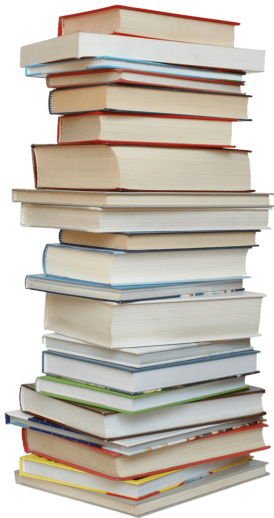
Departamento de Ciência da Computação



Pilha é um outro exemplo de estrutura de dados básica

- Estrutura de dados linear
- Segue a política LIFO (*Last in, First Out*)
 - Último a entrar, primeiro a sair
 - Somente o item do topo é acessível
 - Adições e remoções são sempre realizadas no topo
- Organização LIFO ↑
- Operações em $O(1)$ ↑
- Ótimo para recursão ↑
- Manipulação restrita ao topo da pilha ↓
- Estrutura pouco flexível ↓

Pode ser aplicada em editores, em navegação, em recursão, dentre outros assuntos



REPRESENTAÇÃO DE UMA PILHA

```
1  struct pilha {  
2      int topo;  
3      int tamanho;  
4      int* items;  
5  };
```

REPRESENTAÇÃO DE UMA PILHA

```
1  struct pilha {  
2      int topo;  
3      int tamanho;  
4      int* items;  
5  };
```

topo: Posição do último item da pilha

tamanho: Número de itens que cabem na pilha

items: Vetor contendo os elementos da pilha

OPERAÇÕES POSSÍVEIS COM PILHAS

- `void cria_pilha(<tamanho>);`
 - Cria uma nova pilha
- `void push(<valor>);`
 - Insere item no topo da pilha
- `<tipo> pop(void);`
 - Remove item do topo da pilha
- `bool ehVazia(void);`
 - Verifica se pilha é vazia
- `bool ehCheia(void);`
 - Verifica se pilha é cheia
- `<tipo> obtem_elemento(void);`
 - Obtém o elemento do topo da pilha

`std::stack` [▶ Link](#)

Em C++, uma pilha não tem tamanho máximo

- Utiliza alocação dinâmica de itens
- Ponteiros

Oferece interface para todas as funções do slide anterior

- Exceto *ehCheia*, por motivos óbvios

O código completo da pilha está disponível no Github



► Link

ATIVIDADE PRÁTICA

Modificar o código do Github para implementar pilha igual ao C++

Quer dizer, você tem que usar ponteiros e alocação dinâmica

Fazer com que `int* items` seja uma lista

PRÓXIMA AULA:

FILAS