

# MAC121 - Algoritmos e Estruturas de Dados I

Universidade de São Paulo

Segundo Semestre de 2020

## Pilhas - definição e aplicação

## Problema - parêntesis, colchetes e chaves bem casadas

**Problema:** Dada uma sequência de parêntesis, colchetes e chaves, decida se a sequência é bem formada.

- ▶ ( ) é bem formada;
- ▶ ( ) { ( ) [ ( ) { } ] [ ] } é bem formada;
- ▶ [ ( ] ) **não** é bem formada;
- ▶ ( { } ) ] **não** é bem formada.

Como resolver o problema?

## Pilhas



Uma **pilha** é uma estrutura de dados linear em que só temos acesso ao elemento do topo, como em uma pilha de pratos. Dessa forma, o elemento que pode sair da estrutura é o último que entrou (*Last In First Out*).

## Pilhas

As operações permitidas em uma pilha são:

- ▶ `pilha * criaPilha();`
- ▶ `void destroiPilha (pilha * P);`
- ▶ `void empilha (pilha *P, item x);` (push)
- ▶ `item desempilha (pilha *P);` (pop)
- ▶ `int pilhaVazia (pilha *P);`
- ▶ `item topoDaPilha (pilha *P);`

## Implementando uma pilha

Primeira ideia: Vetor de capacidade fixa

```
#define MAX ...
typedef int item;
typedef struct {
    item v[MAX];
    int topo;
} pilha;
```

Como escrever as funções de pilha?

## Implementando uma pilha

### Segunda ideia: Vetor alocado dinamicamente

```
typedef int item;  
typedef struct {  
    item *v;  
    int topo;  
    int max;  
} pilha;
```

Além das funções de pilha devemos implementar:

▶ `pilha * aumentaPilha (pilha *P);`

## Sequências bem formadas

Voltando ao problema das sequências bem formadas...