Estrutura de Dados Pilha

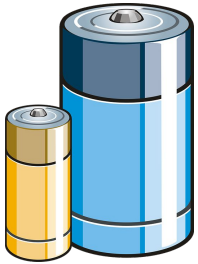
Prof. Joaquim Uchôa
Profa. Juliana Greghi
Prof. Renato Ramos

- Definição e Visão Geral
- Operações Básicas sobre Pilhas
- Informação Importante
- Implementação Utilizando Vetores

Definição e Visão Geral

Definição

- De acordo com o dicionário o que é uma pilha?



Definição

- De acordo com o dicionário o que é uma pilha?[1]
  - Porção de coisas dispostas umas sobre as outras (monte);
  - Dispositivo que transforma a energia química em energia elétrica;
  - Indivíduo irritado ou nervoso
  - Caixa que contém dentro de si outras caixas de várias dimensões, acondicionadas, umas dentro das outras, de acordo com os tamanhos

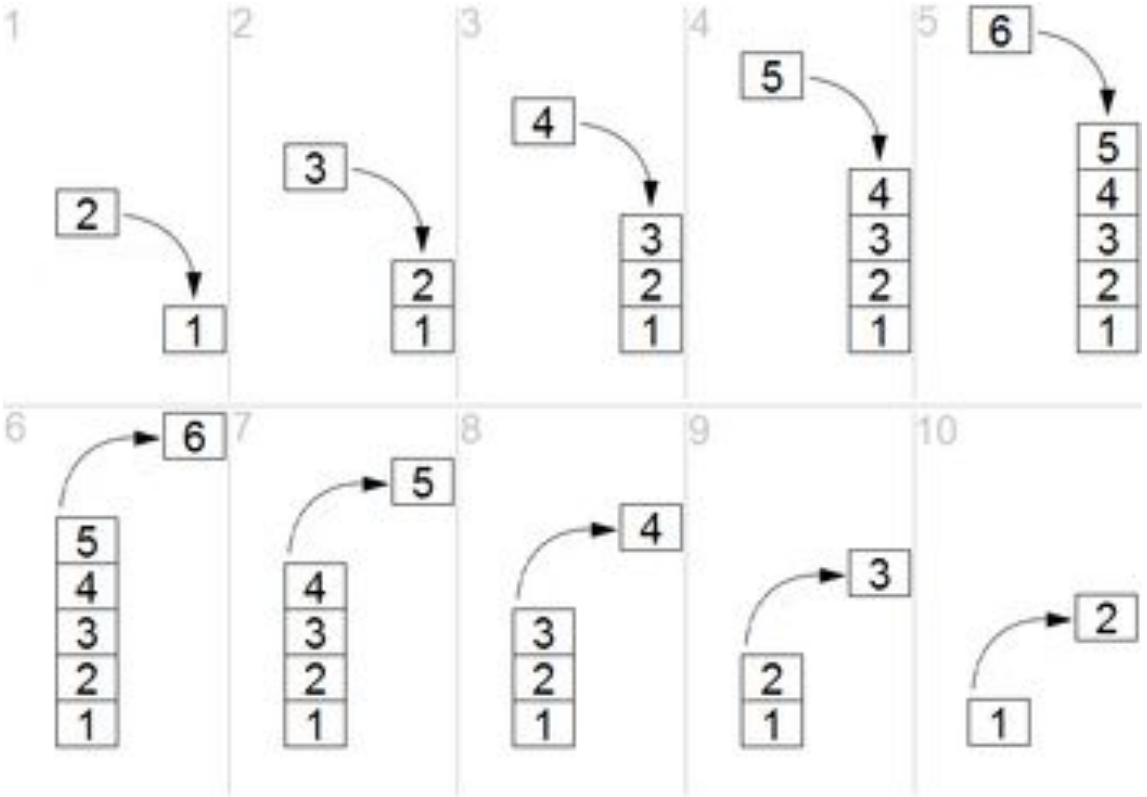
[1]Segundo dicionário michaelis on-line http://michaelis.uol.com.br/busca?id=OWQE

Definição

- E em Computação?
  - É uma estrutura de dados que assume a ideia de monte para armazenar as informações.
  - É baseada no princípio "*Last In First Out*"(LIFO)
    - O último inserido será o primeiro a ser retirado.
    - **A manipulação é feita somente em uma das extremidades.**
    - **Não é possível o acesso direto aos demais dados.**

Definição

Ideia Básica



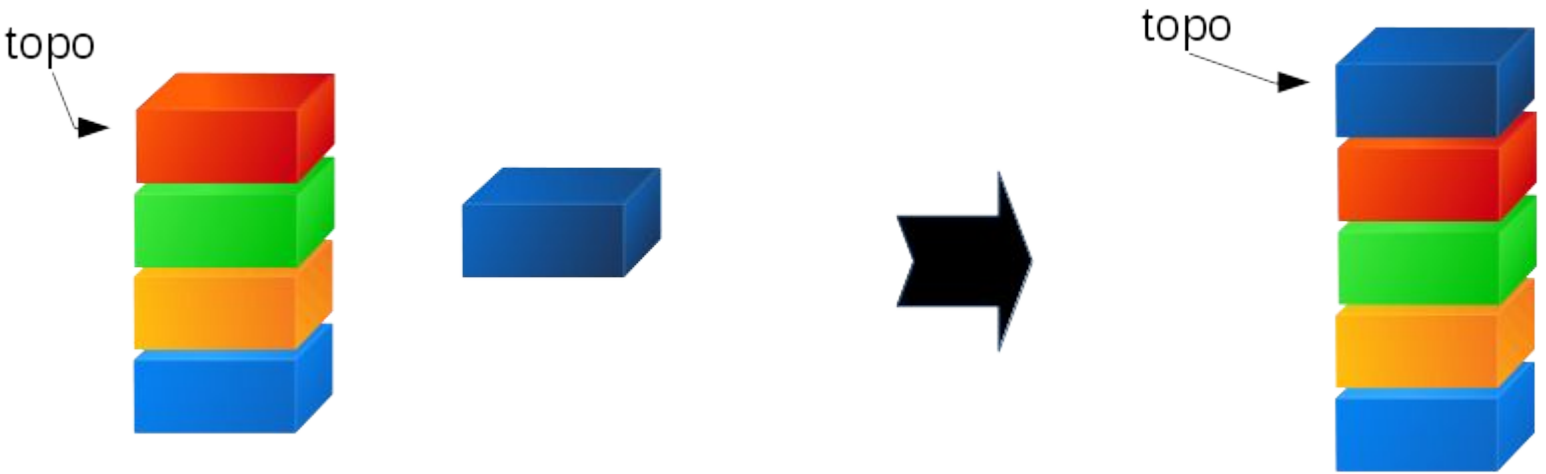
Aplicações Tradicionais de Pilhas

Pilhas são utilizadas em diversas situações na computação. Por exemplo, são utilizadas no **processo recursivo** e nas seguintes aplicações:

- Calculadora para expressões matemáticas;
- Conversão de número decimal para binário;
- Retirada de mercadorias de um caminhão de entregas;
- Mecanismo de fazer/desfazer de editores diversos;
- Mecanismo de navegação de páginas na Internet (avançar e retornar).

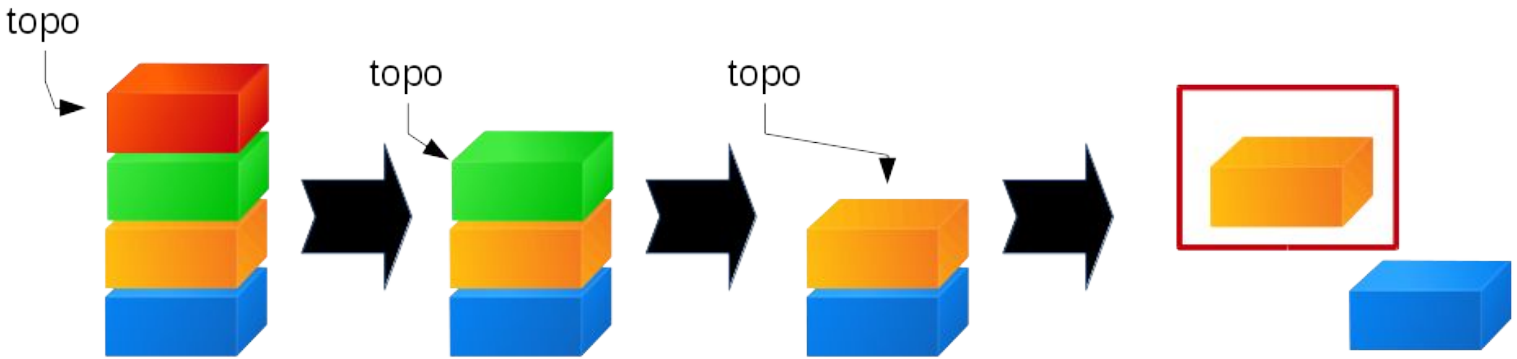
Inserção na Pilha

A inserção deve ser sempre na posição à frente (ou acima) do último elemento inserido.



Retirada de Elementos na Pilha

A retirada de um elemento na pilha é sempre daquele que foi inserido por último. Caso se deseje acessar, por exemplo, o conteúdo de um nó intermediário, todos os demais devem ser desempilhados antes.



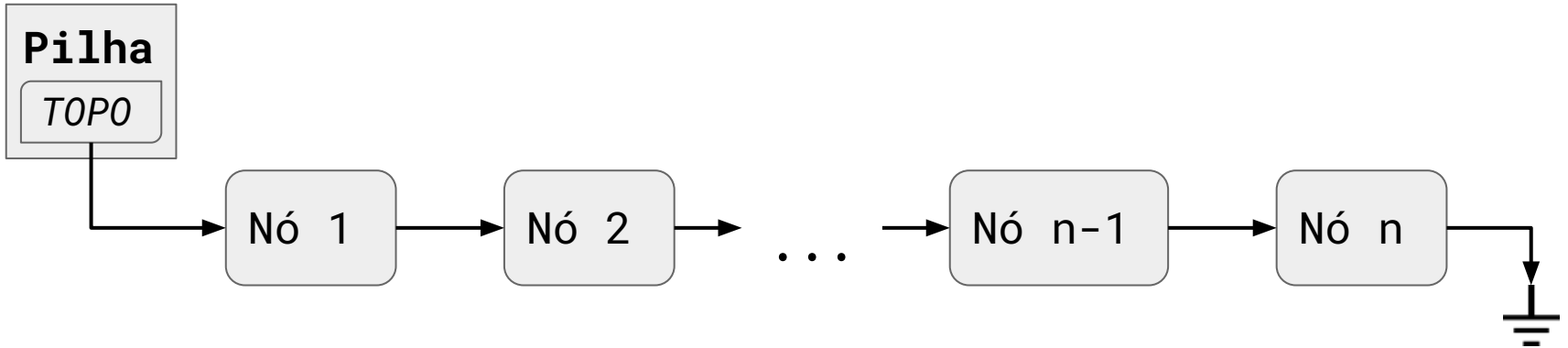
Formas de Implementação

Uma pilha pode ser implementada de diferentes maneiras, as mais comuns são:

- Implementação usando encadeamento de nós (elementos) da pilha;
- Implementação usando arranjo para armazenamento dos elementos.

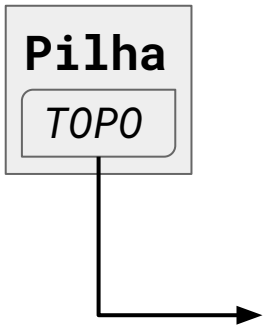
Implementação Usando Encadeamento - I

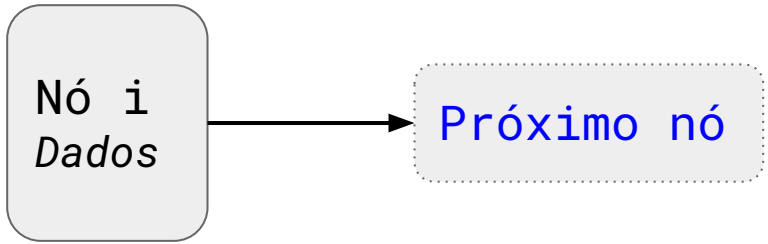
A implementação mais usual de pilha é a usando encadeamento. Nessa implementação, cada nó contém informações e aponta para o próximo nó. O último nó da pilha aponta para NULO, indicando seu término. A pilha sabe apenas onde está o topo.



Implementação Usando Encadeamento - II

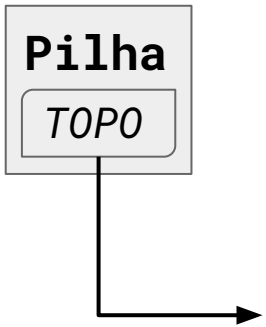
Em uma implementação por encadeamento, uma pilha contém um apontador para o topo e cada nó contém o dado armazenado e um apontador para o próximo nó na estrutura.

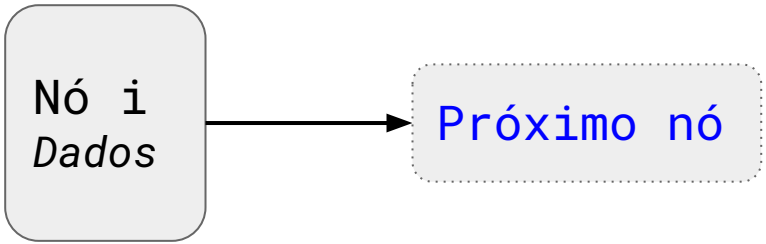




Implementação Usando Encadeamento - III

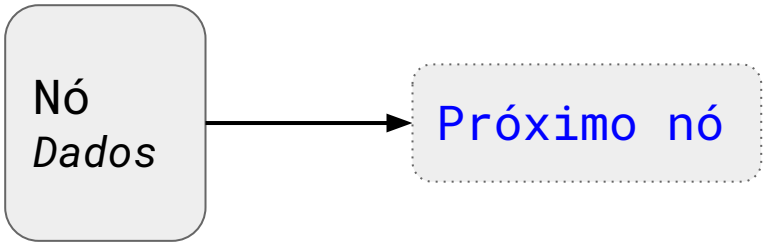
Pilha e nós são implementados tradicionalmente como estruturas ou classes. Usaremos a segunda abordagem, usando uma classe para pilha e outra para os nós.





Classe Noh

Cada nó da pilha possui atributos com os dados armazenados, que pode ser desde um simples inteiro até uma estrutura ou outra classe. Além disso, possui um apontador para o próximo nó.

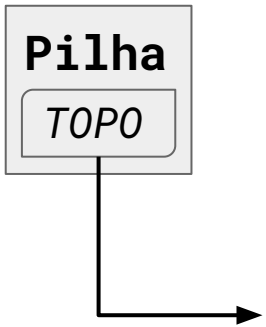


class noh

```
int dado;
noh* proximo;
```

Classe Pilha

A classe pilha contém geralmente, além do apontador para o topo, um atributo para informar a quantidade de elementos (nós) que ela possui. Outros atributos podem ser necessários, dependendo do problema resolvido.



class pilha

```
int tamanho;
noh* topo;
```

Operações Básicas Sobre Pilhas

Operações Básicas sobre Pilhas

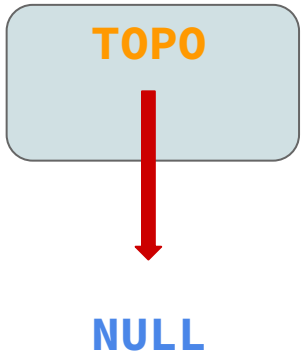
Uma pilha possui um conjunto de operações básicas, para seu correto uso:

- Criação / destruição
- Inserção (ou empilhamento)
- Remoção (ou desempilhamento)

Criação da pilha

criarPilha():

```
topo ← NULL;

// informação opcional
tamanho ← 0;
```



A criação da pilha consiste basicamente em garantir que seu topo não aponte para uma região válida de memória.

Inserção de elemento

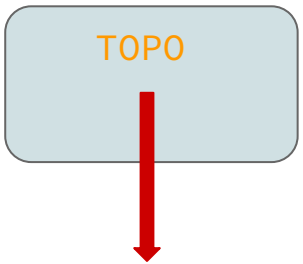
push(valor):

```
novo ← criar_noh(valor);

novo.proximo ← topo;

topo ← novo;

tamanho++;
```

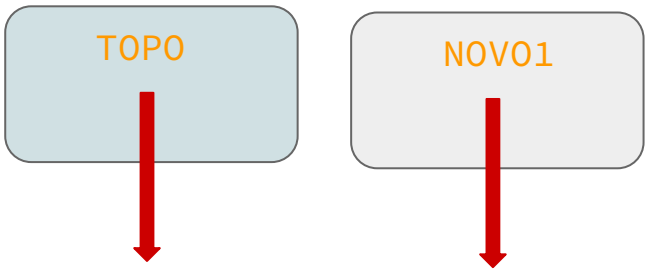
A inserção em uma pilha, consiste em criar um novo nó com o dado a ser armazenado e posteriormente colocá-lo no topo. Assim, seu próximo elemento é o topo antigo e o topo passa a ser o novo elemento inserido.

Operações Básicas Sobre Pilhas

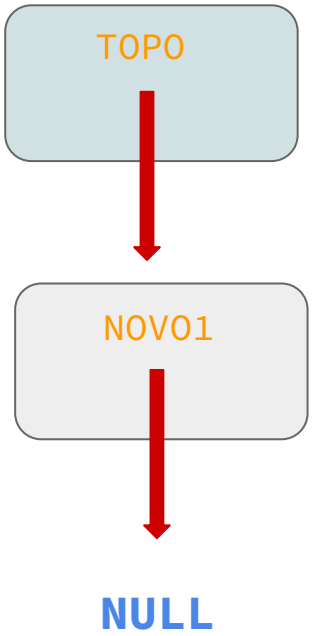
Inserção ou Empilhamento (PUSH)



Situação Inicial

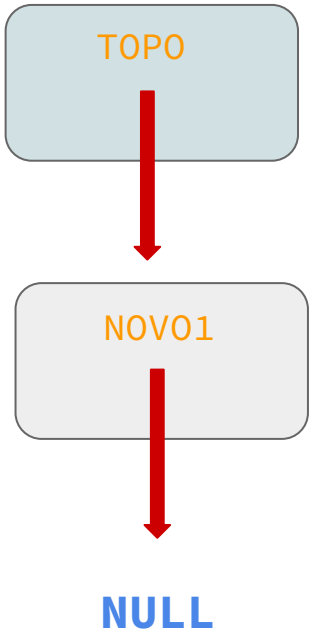


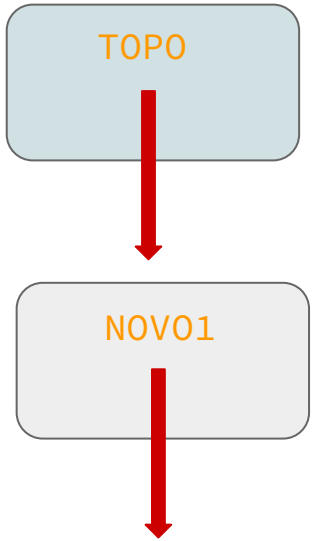
Inserindo o elemento NOVO1

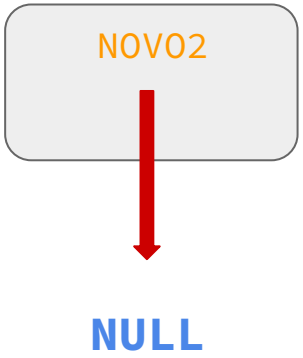


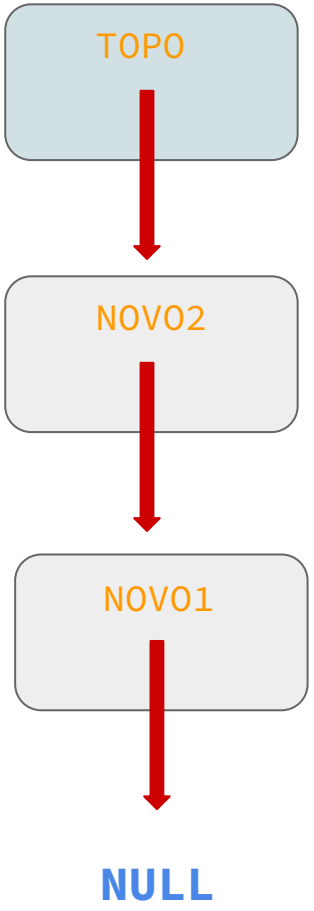
Operações Básicas Sobre Pilhas

Inserção ou Empilhamento (PUSH)







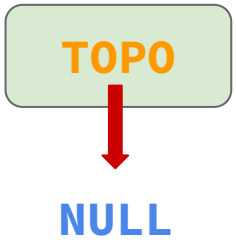


Inserção de elemento

push(valor):

```
novo ← criar_noh(valor);

novo.proximo ← topo;

topo ← novo;

tamanho++;
```

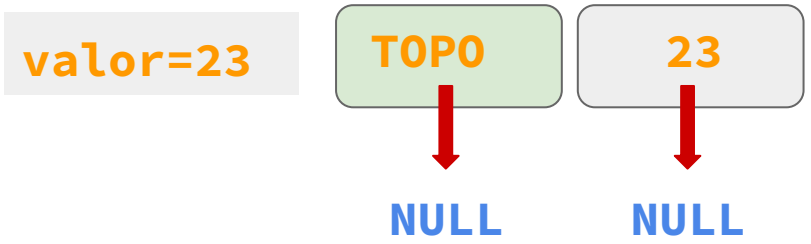
valor=23



Inserção de elemento

push(valor):

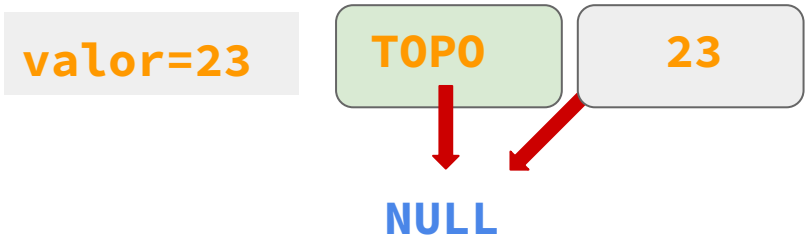
```
novo ← criar_noh(valor);

novo.proximo ← topo;

topo ← novo;

tamanho++;
```



Inserção de elemento

push(valor):

```
novo ← criar_noh(valor);

novo.proximo ← topo;

topo ← novo;

tamanho++;
```

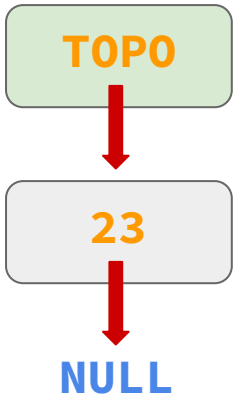


Inserção de elemento

push(valor):

```
novo ← criar_noh(valor);

novo.proximo ← topo;

topo ← novo;

tamanho++;
```

valor=23

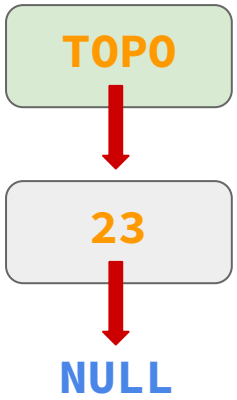


Inserção de elemento

push(valor):

```
novo ← criar_noh(valor);

novo.proximo ← topo;

topo ← novo;

tamanho++;
```

valor=23



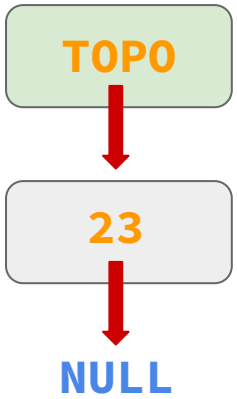
tamanho=1

Inserção de elemento

push(valor):

```
novo ← criar_noh(valor);

novo.proximo ← topo;

topo ← novo;

tamanho++;
```

valor=45

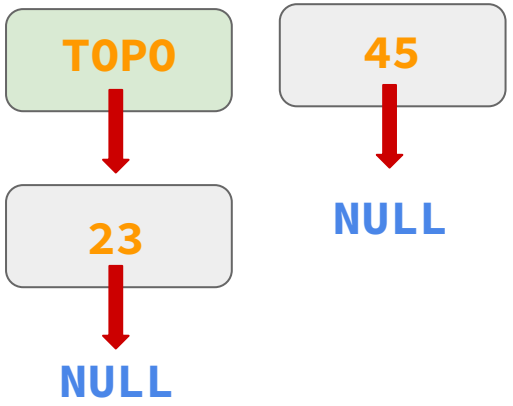


Inserção de elemento

push(valor):

```
novo ← criar_noh(valor);

novo.proximo ← topo;

topo ← novo;

tamanho++;
```

valor=45

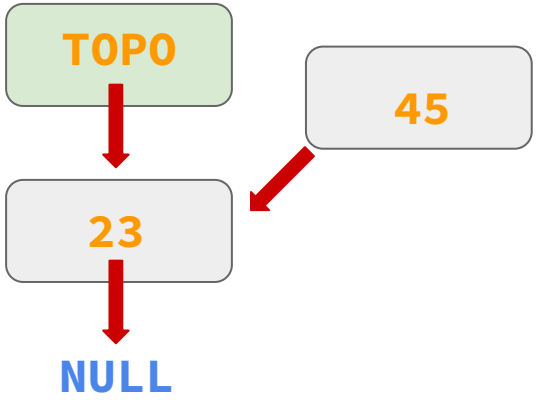


Inserção de elemento

push(valor):

valor=45

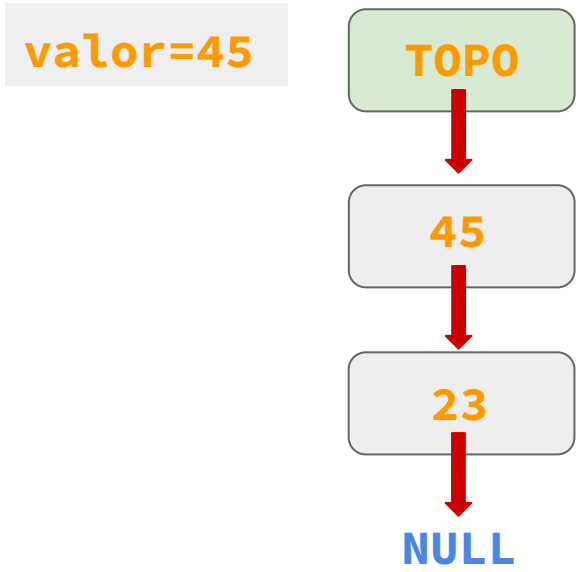
```
novo ← criar_noh(valor);

novo.proximo ← topo;

topo ← novo;

tamanho++;
```



Inserção de elemento

push(valor):

```
novo ← criar_noh(valor);

novo.proximo ← topo;

topo ← novo;

tamanho++;
```

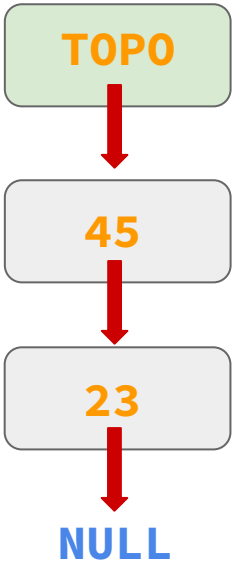


Inserção de elemento

push(valor):

```
novo ← criar_noh(valor);

novo.proximo ← topo;

topo ← novo;

tamanho++;
```

valor=45



tamanho=2

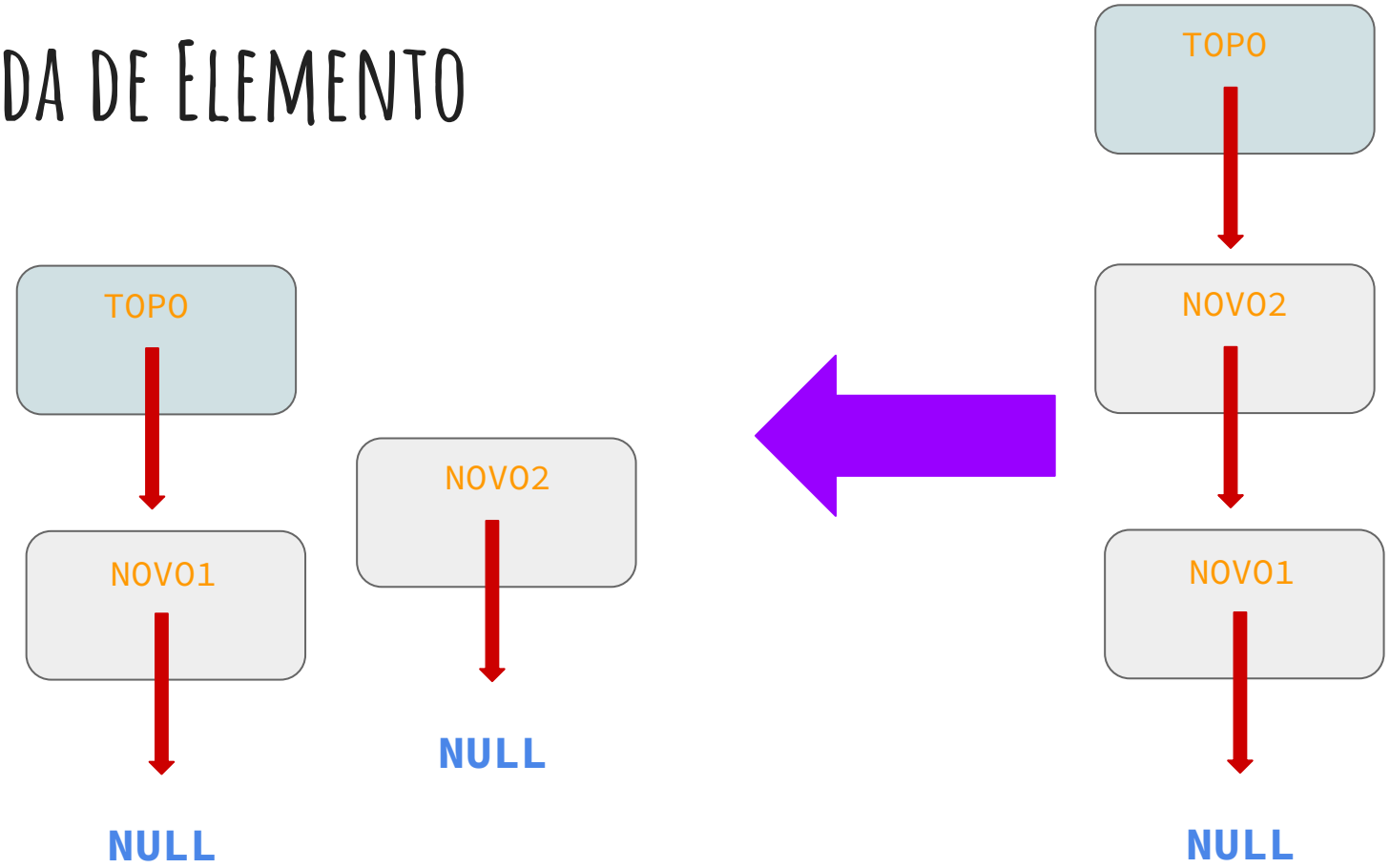
Retirada de Elemento

pop():

```
aux ← topo;
dado ← aux.valor;
topo ← aux.proximo;
apagar(aux);
tamanho--;
// faz ação desejada
// (e.g.: retorno)
efetuaAcao(dado);
```

A retirada de um elemento da pilha consiste em acessar o seu topo, retornando-o para a ação desejada. Nesse caso, o novo topo passa a ser o próximo nó do topo anterior, que será removido do sistema.

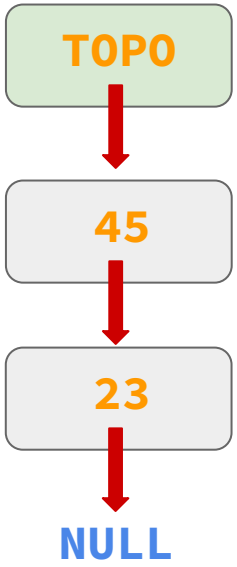
Retirada de Elemento



Retirada de Elemento

pop():

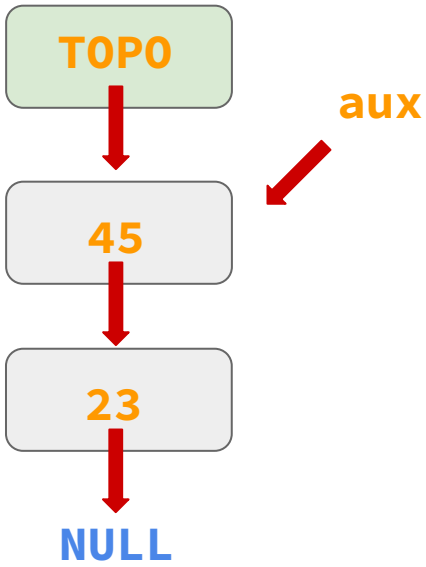
```
aux ← topo;
dado ← aux.valor;
topo ← aux.proximo;
apagar(aux);
tamanho--;
// faz ação desejada
// (e.g.: retorno)
efetuaAcao(dado);
```



Retirada de elemento

pop():

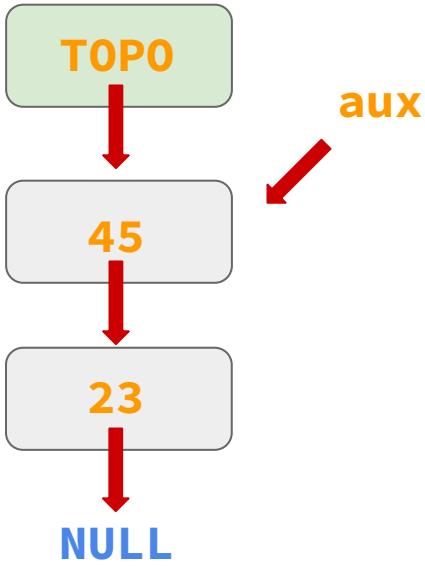
```
aux ←  topo;
dado ← aux.valor;
topo ← aux.proximo;
apagar(aux);
tamanho--;
// faz ação desejada
// (e.g.: retorno)
efetuaAcao(dado);
```



Retirada de elemento

pop():

```
aux ←  topo;
dado ← aux.valor;
topo ← aux.proximo;
apagar(aux);
tamanho--;
// faz ação desejada
// (e.g.: retorno)
efetuaAcao(dado);
```

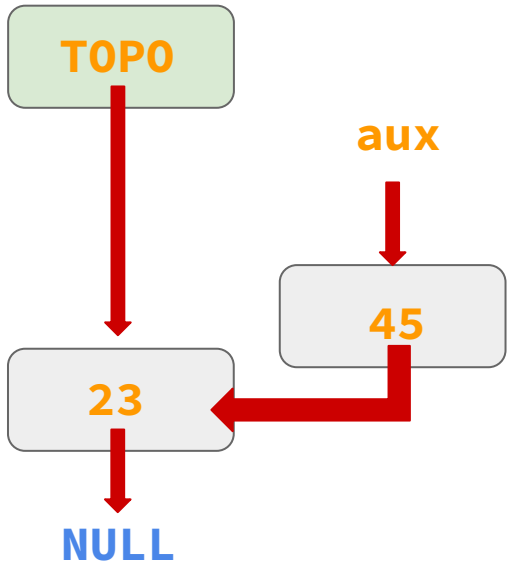


dado=45

Retirada de elemento

pop():

```
aux ←  topo;
dado ← aux.valor;
topo ← aux.proximo;
apagar(aux);
tamanho--;
// faz ação desejada
// (e.g.: retorno)
efetuaAcao(dado);
```

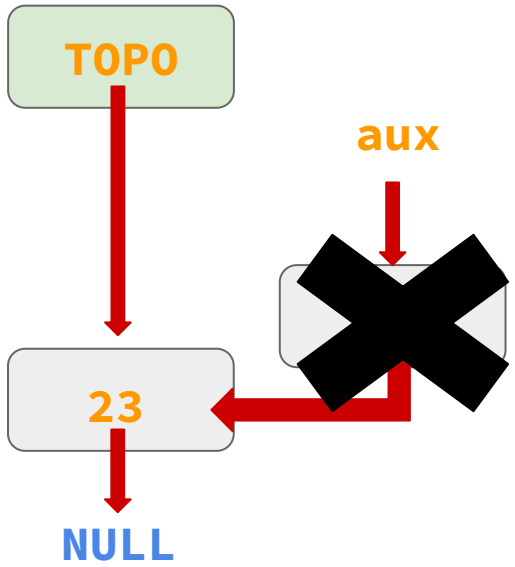


dado=45

Retirada de elemento

pop():

```
aux ← topo;
dado ← aux.valor;
topo ← aux.proximo;
apagar(aux);
tamanho--;
// faz ação desejada
// (e.g.: retorno)
efetuaAcao(dado);
```

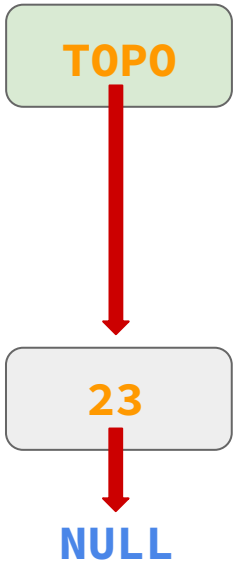


dado=45

Retirada de elemento

pop():

```
aux ←  topo;
dado ← aux.valor;
topo ← aux.proximo;
apagar(aux);
tamanho--;
// faz ação desejada
// (e.g.: retorno)
efetuaAcao(dado);
```

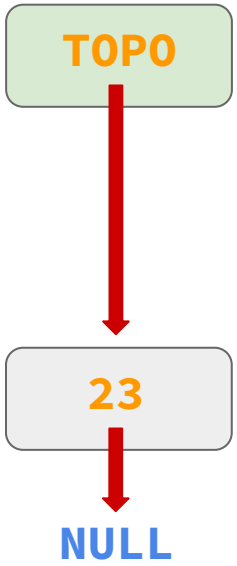



```
dado=45
tamanho=1
```

Retirada de elemento

pop():

```
aux ←  topo;
dado ← aux.valor;
topo ← aux.proximo;
apagar(aux);
tamanho--;
// faz ação desejada
// (e.g.: retorno)
efetuaAcao(dado);
```

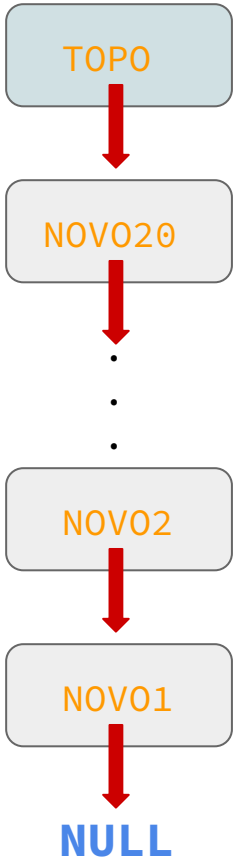



```
dado=45
tamanho=1
```

Informação Importante

Informação Importante

- O acesso à Pilha deve ser sempre feito pelo TOPO
- As ligações entre os nós é para que a estrutura permaneça integrada.



Informação Importante

Mas professor, dessa forma eu só posso acessar o elemento que o TOPO aponta? E se, por exemplo, na estrutura anterior, eu quiser utilizar o elemento NOVO12, o que fazer?

Informação Importante

Pela definição, não é possível acessar os outros elementos que não estão no topo. Assim, não é possível desenvolver uma função da pilha que resolva isso.

Mas você pode retirar todos os elementos do topo até encontrar o elemento desejado. Vamos deixar claro, que essa situação deve ser feita via retirada de elementos, **não** pode ser um procedimento da pilha.

Outras Operações em Pilhas - I

Além das operações básicas apresentadas, as pilhas podem implementar:

- verificar o número de elementos da pilha;
- verificar se a pilha está ou não vazia;
- retirar todos os elementos da pilha;
- opcionalmente acessar o topo da pilha;
- opcionalmente depurar/imprimir a pilha.

Outras Operações em Pilhas - II

- A retirada de todos os elementos da pilha é por meio de desempilhamento, um por um.
- O destrutor é implementado geralmente usando o método para retirar todos os elementos da pilha.

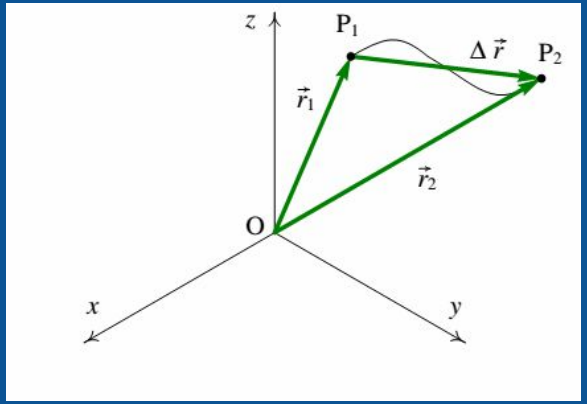
Outras Operações em Pilhas - III

- O acesso ao topo (espia) costuma ser disponibilizado em várias implementações. Nesse caso, há o acesso ao dado, mas sem sua retirada, que continua no topo da pilha.

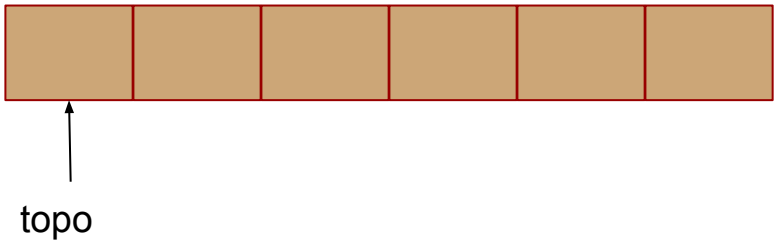
Outras Operações em Pilhas - iv

- Implementações iniciais às vezes fazem uso de mecanismos de percorrimento para impressão/depuração dos dados. Esse método é considerado uma **quebra de estrutura**, pois acessa os elementos intermediários sem desempilhamento. Assim, esse recurso só pode ser usado, para fins didáticos ou de depuração.

Implementação Utilizando Vetor



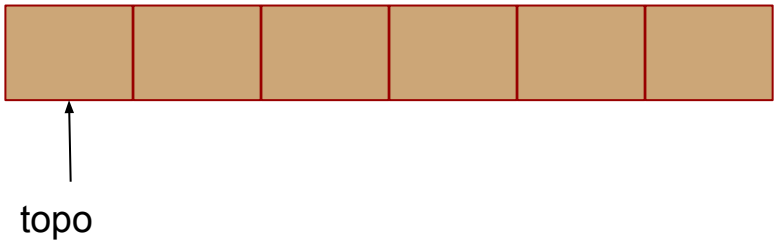
Implementação utilizando vetor



Vantagem

- Operações rápidas (acesso, empilhamento, desempilhamento).

Implementação utilizando vetor

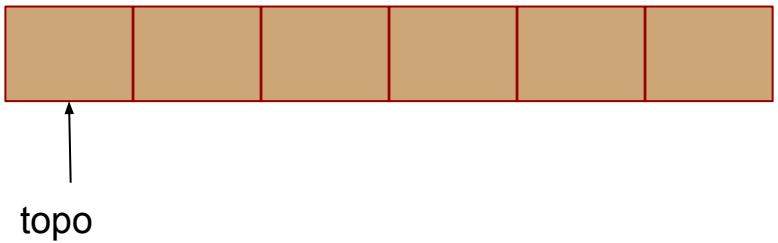


Desvantagem

- Espaço alocado sem necessidade;

- Impossibilidade de aumentar a capacidade da pilha sem redimensionamento do vetor.

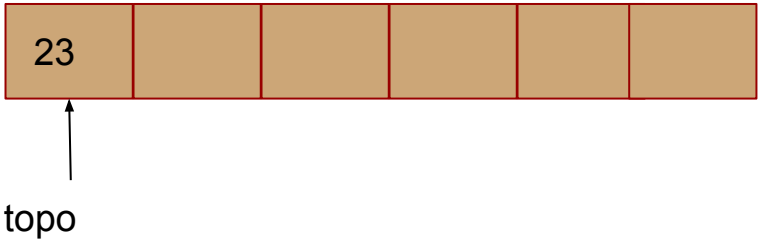
Implementação utilizando vetor



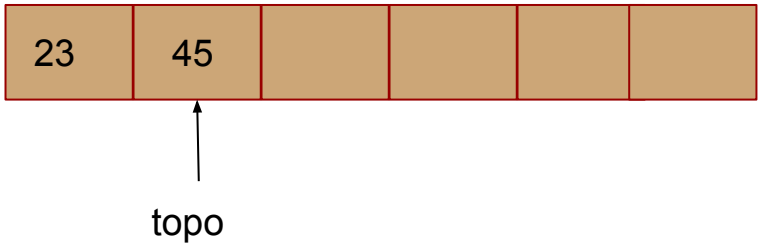
A implementação em vetor utiliza um atributo para armazenar o índice do vetor que contém o topo da pilha. Em uma inserção, o topo avança uma posição. Em uma retirada, o topo retorna uma posição.

Considere os elementos 23, 45, 11 e 89.

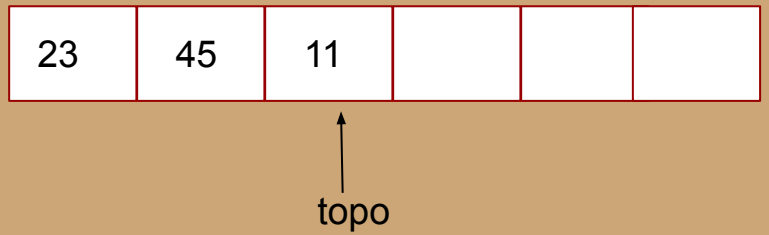
Insere elemento na pilha



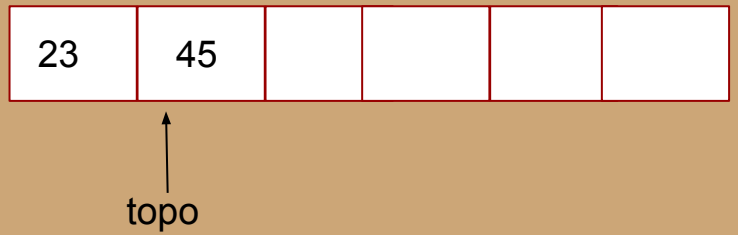
Insere elemento na pilha



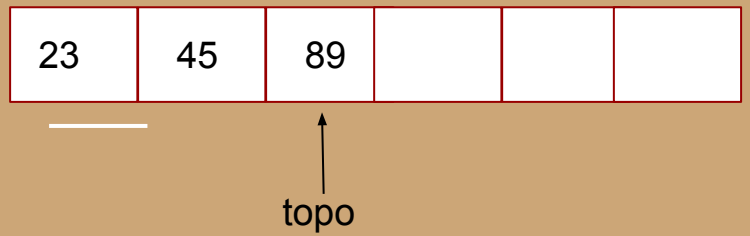
Insere elemento na pilha



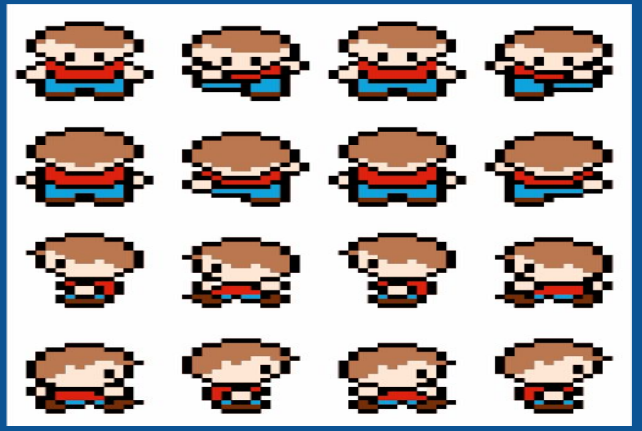
Remove elemento da pilha



Insere elemento na pilha



Um Exemplo: Notação Pósfixa



Expressão em Notação Pósfixa - 1

Um dos problemas ao tratar expressões matemáticas computacionalmente é a precedência de operadores:

2+4*2 = 10? Ou 2+4*2 = 12?

Por conta disso, usamos geralmente parênteses:

2+(4*2) = 10 e (2+4)*2 = 12

Mas internamente, geralmente se usam outras formas de representação.

Expressão em Notação Pósfixa - II

Uma das formas de representar expressões sem o problema com precedência de operadores é o uso de notação pósfixa. Nesse caso, o operador é inserido após os operandos a que deve ser aplicado. Por exemplo:

2+(4*2) seria escrito como 2 4 2 * +

(2+4)*2 seria escrito como 2 4 + 2 *

Expressão em Notação Pósfixa - III

A ideia é que o processamento é feito da seguinte forma: sempre que um operador é lido, ele é aplicado aos dois operandos anteriores. Então vejamos:

2 4 2 * + => ao percorrer, encontramos o *, então ele é aplicado a 4 e 2, produzindo 8. Na sequência, o + é aplicado ao 8 e o 2 inicial, produzindo 10.

Expressão em Notação Pósfixa - iv

2 4 + 2 * => ao percorrer, encontramos o +, então ele é aplicado a 2 e 4, produzindo 6. Na sequência, o * é aplicado ao 6 e o 2 final, produzindo 12.

Usando Pilhas para Expressão em Notação Pósfixa

Uma das vantagens dessa notação é a possibilidade de uso da estrutura de dados pilha para resolver a expressão.

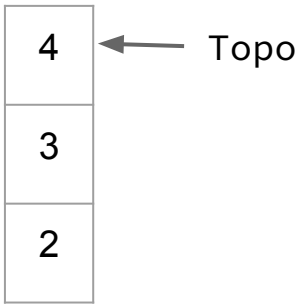
Para isso, basta ir empilhando os valores e, ao encontrar um operador, fazer o desempilhamento dos valores associados.

Exemplo - I

Considere a seguinte expressão pós-fixada:

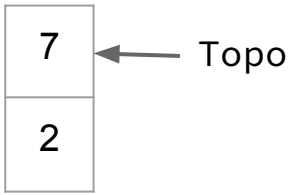
2 3 4 + *

Para calcular o resultado utilizando pilha, empilhamos inicialmente os valores 2, 3 e 4:



Exemplo - II

Ao encontrar o +, o 4 e o 3 são desempilhados e a operação de soma é realizada. O resultado é novamente empilhado:



Por fim, ao encontrar o *, o 7 e o 2 são desempilhados, a operação é realizada e tem-se o resultado final 14, que será o último elemento na pilha.

Sobre o Material

Sobre este material

Material produzido coletivamente, principalmente pelos seguintes professores do DCC/UFLA:

- Joaquim Quinteiro Uchôa
- Juliana Galvani Greghi
- Renato Ramos da Silva

Inclui contribuições de outros professores do setor de Fundamentos de Programação do DCC/UFLA.