



Algoritmo e Programação Pilha

#EU TENHO
PAIXÃO POR EDUCAR

Paulo Rodrigues Milhorato
paulo.rodriques@pitagoras.com.br
Algoritmo e Programação

Indagação inicial

O que é uma pilha?

Citação

“Os últimos serão os primeiros”.

(Evangelho de Mateus, cap. 19, versículo 30)

Introdução

Todos nós já ouvimos esta frase, entretanto muitos não sabem que ela possui um significado na Computação.

Existe uma ideia básica de estrutura de dados presente nela.

As últimas informações guardadas serão as primeiras a serem utilizadas.

Essa característica nos ajudará a resolver alguns problemas computacionais.

Cenário

A pilha é uma estrutura de dados muito frequente em nosso dia a dia.

└ Por isso é de fácil compreensão.

Se você já se mudou de residência ou teve que arrumar algum cômodo, já se deparou com uma pilha.

Vamos considerar o cenário da mudança.

Imagine que você irá finalmente ter uma estante “de respeito” para guardar os seus quadrinhos favoritos.

Cenário

Você recebeu uma caixa de papelão para colocar todos os seus quadrinhos e depois liberá-lo para o caminhão.

Assim, dentro da caixa, um quadrinho ficará sobre o outro quadrinho.

Afinal, você é bem zeloso e não deseja que fique em pé, pois pode amassar as quinas e bordas.

Temos a caixa como a memória do computador e os quadrinhos como as informações que queremos armazenar.

Temos, então, algumas perguntas:

⇒ Como ocorre esse acondicionamento dos quadrinhos nas caixas fornecidas?

⇒ Quais passos você seguiria para armazená-los?

Cenário

O primeiro passo seria escolher uma das prateleiras e escolher a ordem de retirada para a caixa.

O segundo seria o processo de retirar, um a um, os quadrinhos da estante e colocá-los na caixa.

O primeiro quadrinho será colocado no fundo da caixa, o segundo em cima do anterior e assim sucessivamente.

Cenário

O processo de guardar quadrinhos é hipoteticamente fixo.

Isso significa que um quadrinho é selecionado na estante e inserido no topo da pilha dentro da caixa.

Dentro da caixa, sempre que um quadrinho entra, este se torna o topo atual, até que um novo entre.

Este processo ocorrerá enquanto a caixa estiver com espaço.

Cenário

Terminada a mudança, a nova estante estará pronta para receber os quadrinhos que foram transportados em caixas.

Para isto, é preciso esvaziar as caixas!

Depois de escolher uma caixa e abri-la, os quadrinhos serão retirados um a um para alocação na nova estante.

O procedimento é retirar um quadrinho do topo da pilha e alocá-lo em algum local da estante.

Cenário

Este passo será executado para todos os livros, até que a caixa em questão esteja vazia.

Se observamos o processo como um todo, desde a criação da pilha até o esvaziamento dela, observamos:

- ⇒ o primeiro livro inserido foi o último a ser removido, e
- ⇒ o último livro a ser inserido foi o primeiro a ser removido.

Este é o padrão de comportamento em uma pilha!

Pilha

Um pilha sempre será LIFO (Last-In First-Out), ou seja, o último a entrar é o primeiro a sair.

Em nosso exemplo:

- ⇒ a caixa interpretou o papel de memória, e
- ⇒ os quadrinhos interpretaram o papel de informações.

Na Computação temos a memória armazenando qualquer tipo de informação.

São aceitos quaisquer tipos primitivos e também os abstratos.

Pilha

Antes de iniciar as implementações de funções, é preciso ressaltar a importância do tamanho da pilha.

Pilhas possuem um tamanho máximo.

- └ Mesmo que este tamanho seja grande.

Já ouviram falar no erro Stack Overflow?

- └ Tem até um site de perguntas e respostas de programação com este nome.

Pilha

É necessário, portanto, definir um tamanho máximo para a pilha.

└ Pode ser definida como uma constante global, por exemplo.



Evolução do topo da pilha durante as inserções e remoções

A partir deste momento, não falaremos mais em inserir e remover informações na pilha.

Utilizaremos o nome técnico para isto:

- ⇒ inserir na pilha será chamado de push, e
- ⇒ remover algo da pilha será chamado de pop.

Vamos verificar primeiro como insere (push) e como remove (pop).

Pilha – Push

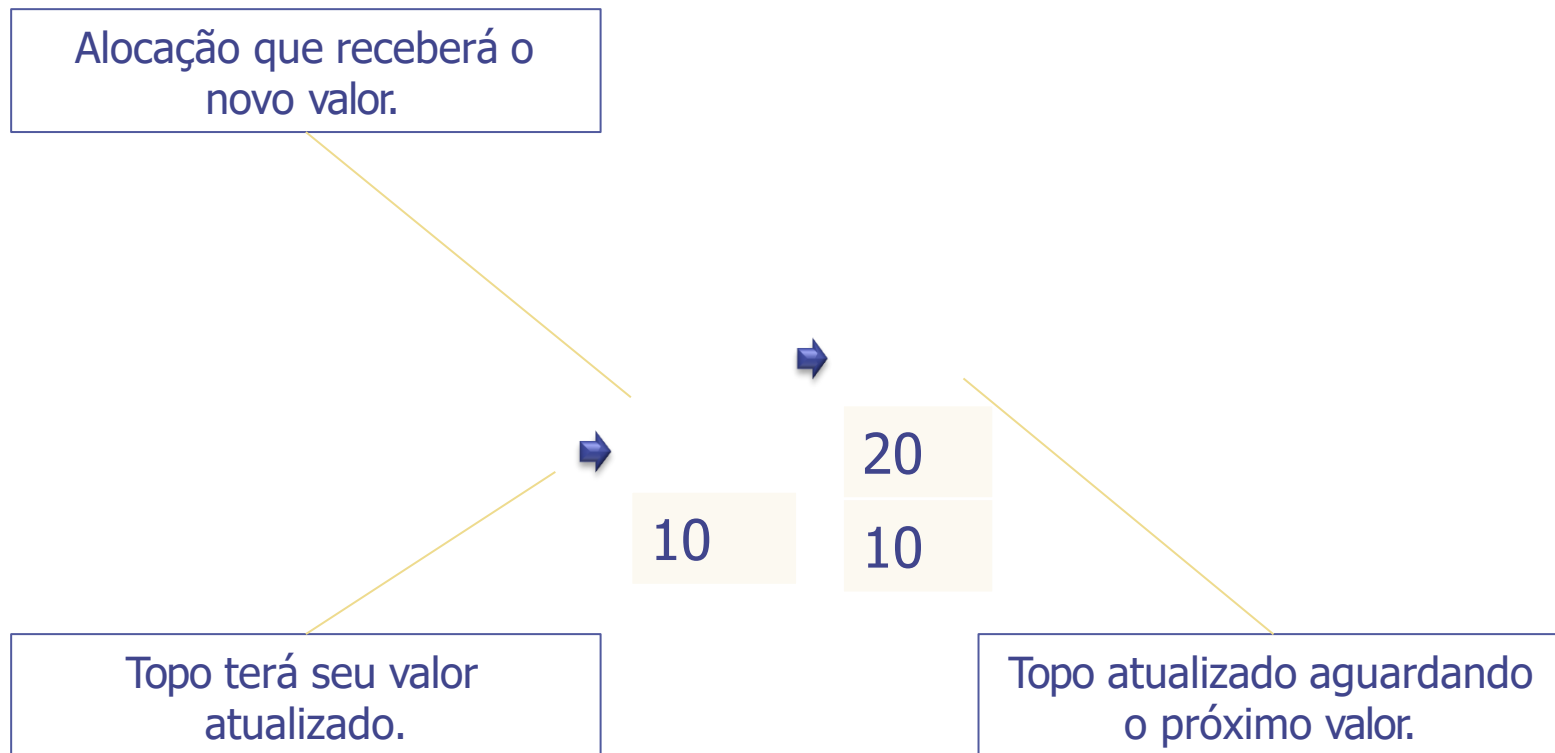


Ilustração dos passos para inserção (push) de um novo elemento na pilha.

Pilha – Pop

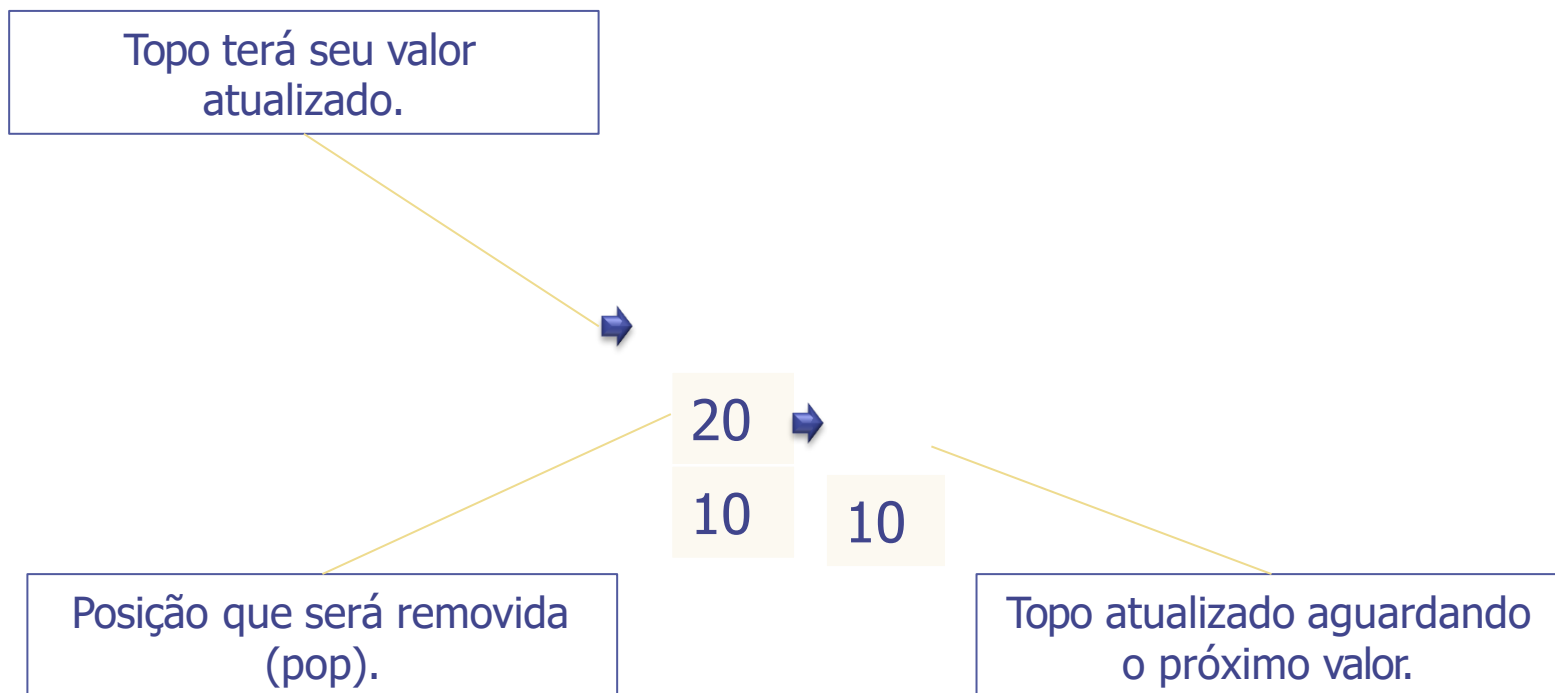


Ilustração dos passos para remoção (pop) de um novo elemento na pilha.

Pilha – Inicializar

Inicializar uma pilha

⇒ Uma pilha vazia significa uma pilha sem informações inseridas.

⇒ Os ponteiros pilha e topo devem apontar para NULL.

⇒ O tamanho da pilha é iniciado com 0.

Pilha – Push

Inserir em uma pilha

⇒ Verificar se a pilha está vazia.

⇒ Se estiver, pilha e topo apontarão para o novo elemento inserido.

⇒ Se não estiver, deve ser inserido ao final, sendo que o antigo último será o penúltimo e o novo elemento o último.

Pilha – Imprimir

Exibir elementos de uma pilha

- ⇒ Verificar se a pilha está vazia.
- ⇒ Se estiver, não há elementos a serem impressos.
- ⇒ Se não estiver, devem ser exibidos os elementos.

OBS: Não existe função de impressão em pilha, uma vez que o último elemento é o único a ser visto.

Pilha – Pop

Remover elementos de uma pilha

⇒ Verificar se a pilha está vazia.

⇒ Se estiver, não há como remover .

⇒ Se não estiver, verificar se há apenas um, para voltar com a pilha para o estado vazia.

⇒ Se possuir 2 ou mais elementos, remover o último, o penúltimo será o último, o antepenúltimo será o penúltimo e assim por diante.

Praticando

Faça um algoritmo que:

- Por meio de um menu faça:
 - Inserção dos dados da pilha
 - Validação se existe um determinado elemento na pilha
 - Imprima toda a pilha
 - Opção para chamar a função abaixo:
- Crie uma função que zere toda a pilha, ou seja, faça pop de todos os elementos.



www.faculdadepitagoras.com.br



Referências

(Cap. 3-4) EDELWEISS, Nina; GALANTE, Eduarda. **Estrutura de dados**. Porto Alegre: Bookman, 2009.

(Cap. 5-6) KERNIGHAN, B. W; RITCHIE, D. M. **The C programming language**. 2. ed. New Jersey: Prentice Hall, 1988.

(Cap. 10) PIVA JUNIOR, D. et al. **Estrutura de dados e técnicas de programação**. Rio de Janeiro: Elsevier, 2014.

(Cap. 20) SCHILDT, Herbert. **C completo e total**. 3. ed. Tradução de Roberto Carlos Mayer. São Paulo: Makron Books, 1996. 810 p.