



Introdução: Estruturas Lineares

As estruturas lineares são aquelas em que todos os dados estão acondicionados em uma fileira sequencial única.

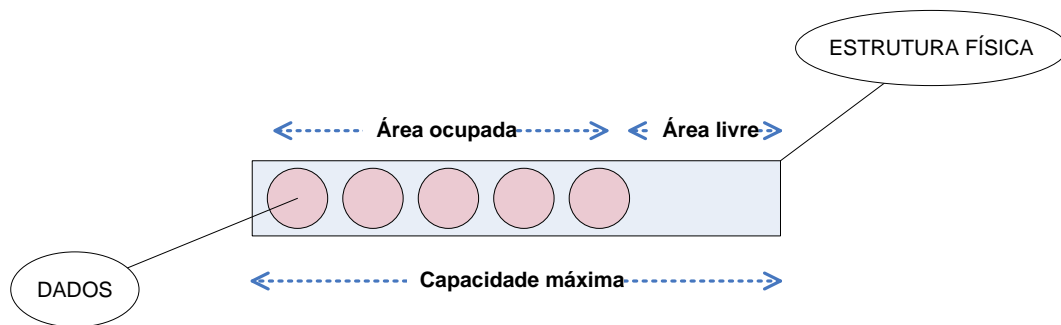


Figura 1

Todas as estruturas de dados pressupõem a possibilidade das operações CRUD (create, read, update e delete), mas, dependendo da natureza do problema a ser representado, algumas operações podem ser inibidas e outras podem ser acrescentadas.

- “Create” implica em inserir um novo elemento na estrutura.
- “Delete”, em extrair um elemento da estrutura.
- “Update”, em alterar o valor de um elemento na estrutura.
- “Read” tem duas conotações:
 1. Ler o valor de um elemento em determinada posição previamente conhecida.
 2. Procurar, na estrutura, um ou mais elementos que atendam a determinadas condições.

Algumas estruturas possibilitam a inserção e a remoção apenas pelas suas extremidades. É como se fosse um tubo que, para se obter um objeto que está no meio é necessário retirar primeiramente as que estão na sua frente. Outras já possibilitam a inserção/remoção em qualquer ponto da estrutura.

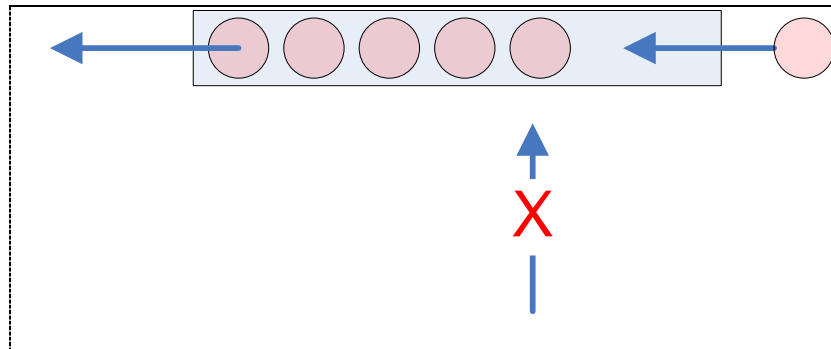


Figura 2 – Acesso apenas pelas extremidades

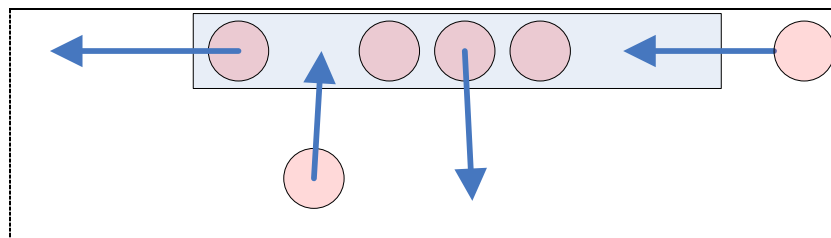


Figura 3 – Acesso pelas extremidades e pelas laterais

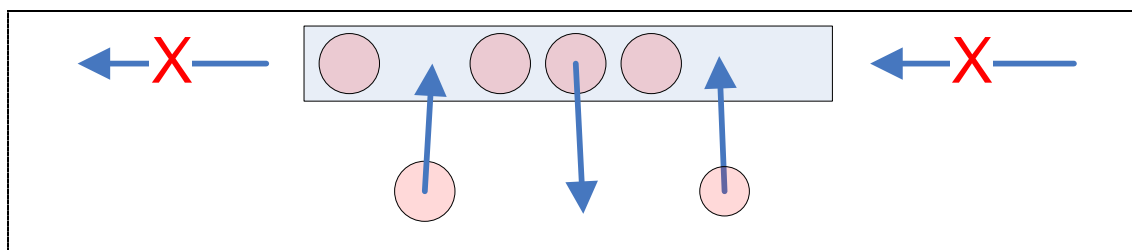


Figura 4 – Acesso apenas pelas laterais

Devemos deixar claro que essa ou aquela “possibilidade” pode ser fisicamente possível ou não. Mas, em função das regras de negócio do problema a ser modelado, mesmo que a estrutura física possibilite determinado acesso, este deverá ser bloqueado pelo programador caso as regras não o permitam.



TAD PILHA

O TAD Pilha (*stack*, no inglês) é uma estrutura linear que permite a inserção/extração dos dados apenas por uma das extremidades, chamada **topo**.

O objetivo do TAD Pilha é representar problemas que envolvam empilhamento de elementos. Ou seja, uma pilha como a conhecemos no mundo real.



Figura 5 - Pilha de caixas



Figura 6 - Pilhas de camisas



Figura 7 - Pilha de carrinhos

Modo de operação

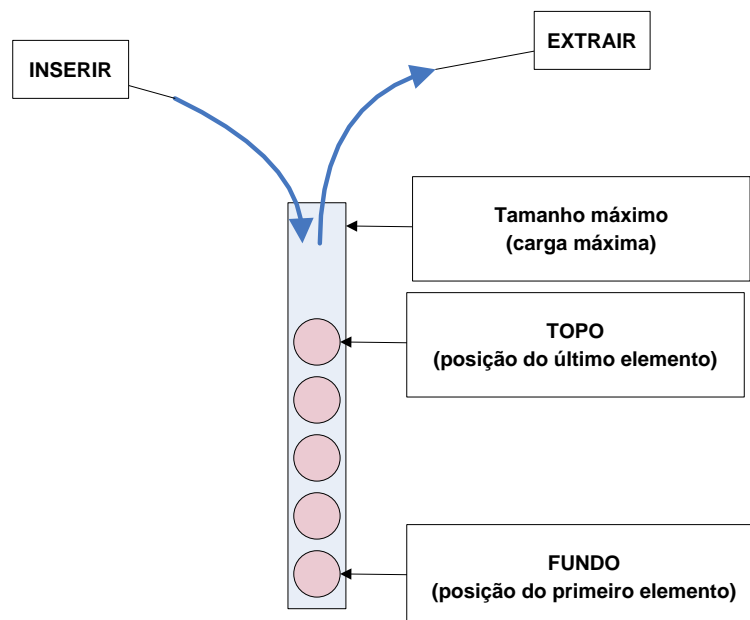


Figura 8

Professor *Marcio Feitosa*



Como em uma pilha real, inserem-se os elementos pelo topo e, também, retiram-se pelo topo. É o modelo chamado LIFO (*last in first out*), ou, traduzindo, o último a entrar será o primeiro a sair.

A retirada ou inserção de um elemento no meio da pilha requer algum tipo de mecanismo que levante os elementos que ficarão acima desse novo para que possa ser encaixado. O TAD Pilha convencional não prevê esta operação.

Computacionalmente falando, precisamos de pelo menos dois métodos no TAD Pilha:

- Para a operação de inserção - `push(elemento)`
- Para a operação de extração - `pop()`

Visibilidade dos elementos

Há aplicações em que os elementos da pilha não podem ser visíveis para o usuário, como por exemplo num jogo de cartas em que há uma pilha de cartas a serem retiradas. Se no jogo real as cartas ficarem viradas e de identidade não visível, um game desse jogo também não poderá permitir a visualização. Somente com a retirada da carta de cima é que sua identidade passará a ser conhecida.

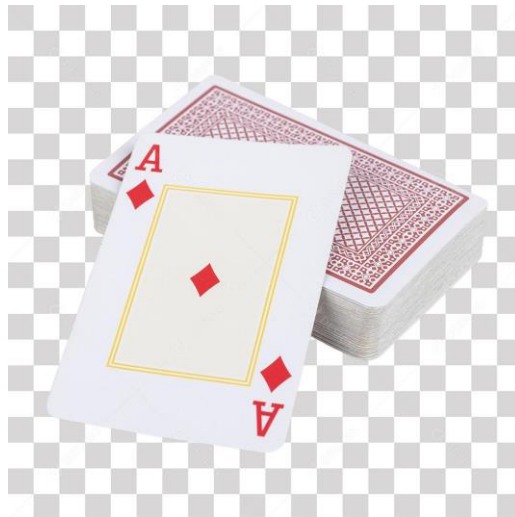


Figura 9 – Pilha de cartas de baralho

Outras aplicações já requerem que a identidade dos elementos seja visível. Por exemplo, o quebra-cabeças **Torres de Hanoi** e o **FreeCell** (*solitaire* de cartas).

Professor Marcio Feitosa

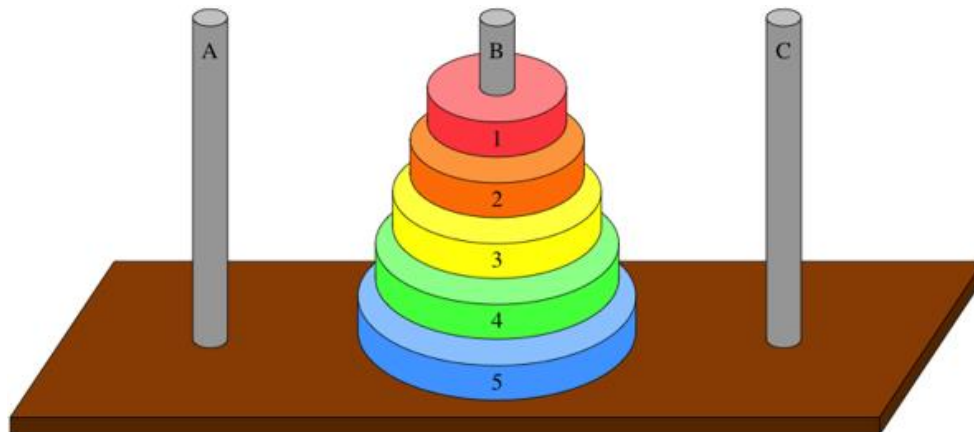


Figura 10 - Quebra-cabeças Torres de Hanoi

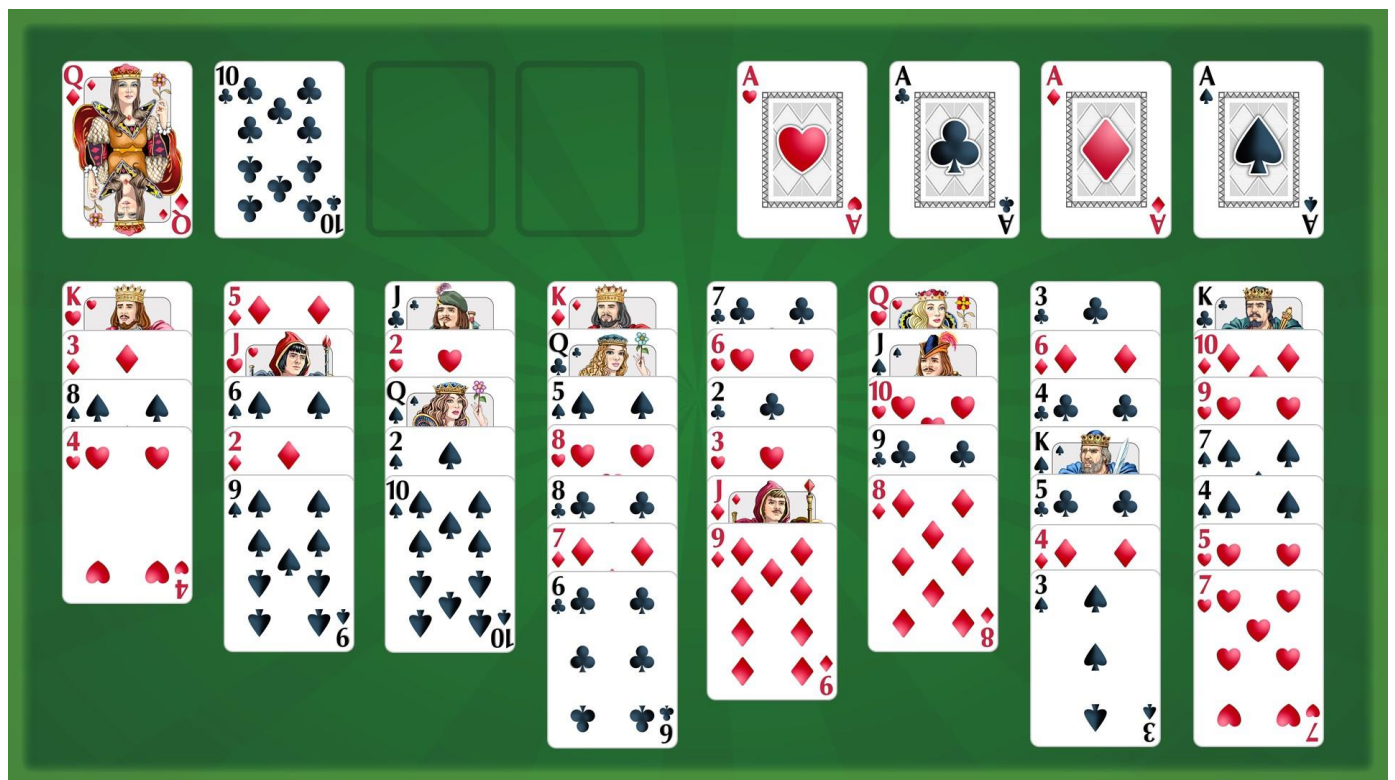


Figura 11 - Jogo de Paciência (Solitaire) - FreeCell

Mas, vamos ao código!!

----- 10101010101010 -----