Pilhas, Filas e Deques

Rafael Alves da Costa

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
FATEC Carapicuíba

Aula 5 - Estrutura de dados

09/2025

Sumário

1 Introdução: Estruturas lineares

2 Pilhas (Stacks)

3 Filas (Queues)

4 Deques

Introdução: Estruturas lineares

Introdução: Estruturas lineares

- Agora começamos nossos estudos de estrutura de dados considerando estruturas simples como, **Pilhas** e **Filas** que são exemplos de coleções de dados cujos itens são ordenados de acordo com ordem que são inseridos ou removidos da estrutura[1].
- Uma vez que um item é inserido, fica em uma mesma posição em relação aos demais itens que foram inseridos antes ou que serão inseridos depois. Essas coleções são frequentemente chamadas de **estruturas de dados lineares**[1].
- O que distingue uma estrutura linear de outra é a maneira em que itens são **inseridos** e **removidos**, em particular a extremidade onde estes inserções e remoções ocorrem. Por exemplo, uma estrutura pode permitir que novos itens sejam inseridos em apenas uma das extremidades[1].

[1] Miller, B. N.; Ranum, D. L. Problem solving with algorithms and data structures using python, 2Ed. Franklin, Beedle and Associates Inc., 2011.

Comparativo de Estruturas Lineares

- Lista Simplesmente Encadeada (LSE)
- Lista Duplamente Encadeada (LDE)

Estrutura	**Acesso**	**Inserção/Remoção**	**Memória**	**Direção**
Array	O(1)	O(n)	Alta	Unidirecional
LSE	O(n)	O(1) no início, O(n) no final	Média	Unidirecional
LDE	O(n)	O(1) se nó for conhecido	Alta (2 pont.)	Bidirecional

Comparativo entre diferentes estruturas lineares que já vimos!

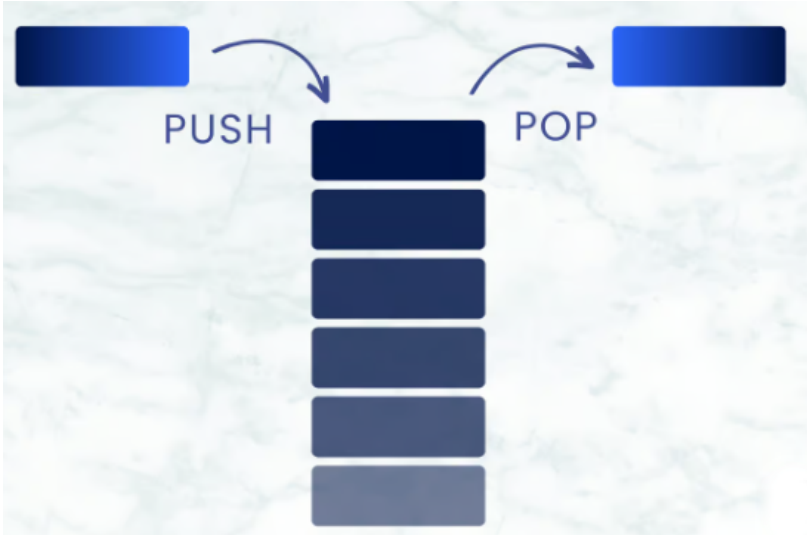
Tipos de Estruturas de Dados

Tipo	**Estruturas**	**Acesso**
Lineares	Array, LSE, LDE, **Pilha, Fila, Deques**	Sequencial ou Direto (Array)
Não Lineares	Árvore, Grafo, Tabela Hash, Heaps	Hierárquico ou por Conexões

Pilhas (Stacks)

Pilhas (Stacks)

- Uma **pilha** é uma coleção de objetos que são inseridos e removidos de acordo com o **princípio do último a entrar, primeiro a sair** (**LIFO** - *Last In, First Out*). Um usuário pode inserir objetos em uma pilha a qualquer momento, mas só pode acessar ou remover o objeto inserido mais recentemente que permanecer (no chamado "topo"da pilha)[2].



  - As **pilhas** são as mais **simples** de todas as **estruturas de dados**, mas também estão entre as mais importantes. Elas são usadas em uma série de aplicativos diferentes e como ferramenta para muitas estruturas de dados e algoritmos mais sofisticados[2].

[2]Goodrich, M. T., Tamassia, R., Goldwasser, M. H. (2013). Data structures and algorithms in Python.

Pilhas (Stacks)

- **Exemplo:** Formalmente, uma pilha é um tipo de dados abstrato (TDA) em que uma instância S suporta os dois métodos a seguir[2]:
  - S.**push(x):** Adiciona x ao topo da pilha S
  - S.**pop():** Remove o elemento do topo da pilha S
  - Adicionalmente pode ocorrer algum método acessório.

Operation	Return Value	Stack Contents
S.push(5)	–	[5]
S.push(3)	–	[5, 3]
len(S)	2	[5, 3]
S.pop( )	3	[5]
S.is_empty( )	False	[5]
S.pop( )	5	[ ]
S.is_empty( )	True	[ ]
S.pop( )	"error"	[ ]
S.push(7)	–	[7]
S.push(9)	–	[7, 9]
S.top( )	9	[7, 9]
S.push(4)	–	[7, 9, 4]
len(S)	3	[7, 9, 4]
S.pop( )	4	[7, 9]
S.push(6)	–	[7, 9, 6]
S.push(8)	–	[7, 9, 6, 8]
S.pop( )	8	[7, 9, 6]

- **Implementação em Python no notebook!!!**

Similaridades entre Pilhas em Java e Python

Conceito LIFO (Last In, First Out)

- Em ambas as linguagens, a lógica fundamental da pilha é a mesma: o último elemento inserido (**push**) é o primeiro a ser removido (**pop**).

Operações Principais

- **push (empilhar)**: Insere um novo elemento no topo da pilha.
- **pop (desempilhar)**: Remove o elemento do topo da pilha.
- **peek (ou top)**: Retorna (sem remover) o elemento do topo da pilha.
- **isEmpty**: Verifica se a pilha está vazia.

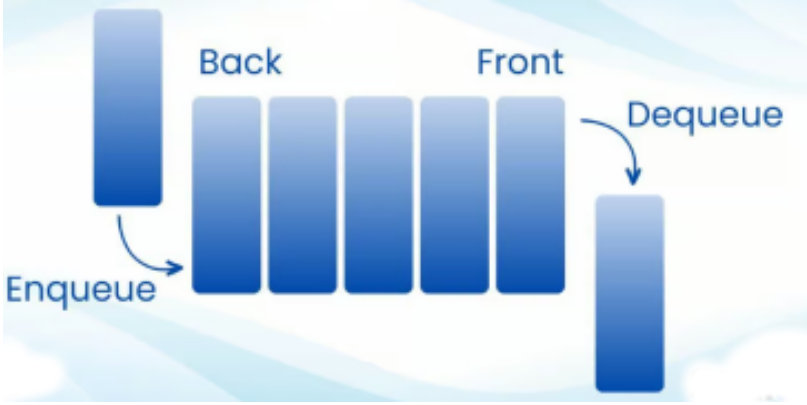
Diferenças entre Pilhas em Java e Python

- **Java** possui a classe `Stack` na biblioteca padrão (chamada `java.util.Stack Class`), mas, na prática atual, recomenda-se o uso de `Deque` (por exemplo, `ArrayDeque`) como pilha.
- **Python** não tem uma classe `Stack` explícita na biblioteca padrão. Usa-se lista (`list`) ou `collections.deque`.
- **Concorrência**: Java oferece classes e interfaces específicas para manipulação segura em threads, enquanto Python conta com o GIL e depende de ferramentas adicionais para maior controle de sincronização.
- **Sintaxe**: Python tende a ser mais enxuto na criação e manipulação de pilhas, ao passo que Java exige definições de tipos e importações de classes específicas.

Filas (Queues)

Filas (Queues)

- Outra estrutura de dados fundamental é a **fila**. Ela é uma "*prima*" próxima da pilha, pois a fila é uma coleção de objetos que são inseridos e removidos de acordo com o **princípio primeiro a entrar, primeiro a sair** (**FIFO** (*first-in, first-out*)). Ou seja, os elementos podem ser inseridos a qualquer momento, mas somente o elemento que estiver na fila há mais tempo poderá ser removido em seguida[2].



  - O TDA fila define uma coleção que mantém objetos em uma **sequência**, em que o **acesso** e a **exclusão** de elementos são restritos ao **primeiro** elemento da fila e a inserção de elementos é restrita ao **final** da sequência[2].

Filas (Queues)

- **Exemplo:** Formalmente, uma fila é um TDA em que uma instância `Q` suporta os dois métodos a seguir[2]:
  - `Q`**.enqueue(x):** Adicionar o elemento x ao final da fila `Q`
  - `Q`**.dequeue():** Remover e retornar o primeiro elemento da fila `Q`
  - Adicionalmente pode ocorrer algum método acessório.

Operation	Return Value	first ← Q ← last
Q.enqueue(5)	–	[5]
Q.enqueue(3)	–	[5, 3]
len(Q)	2	[5, 3]
Q.dequeue( )	5	[3]
Q.is_empty( )	False	[3]
Q.dequeue( )	3	[ ]
Q.is_empty( )	True	[ ]
Q.dequeue( )	"error"	[ ]
Q.enqueue(7)	–	[7]
Q.enqueue(9)	–	[7, 9]
Q.first( )	7	[7, 9]
Q.enqueue(4)	–	[7, 9, 4]
len(Q)	3	[7, 9, 4]
Q.dequeue( )	7	[9, 4]

- **Implementação no notebook em Python!!!**

Similaridades entre Filas em Java e Python

- **Conceito FIFO (First In, First Out):** Em ambas as linguagens, a lógica fundamental de uma fila é a mesma: o primeiro elemento inserido (enqueue) é o primeiro a ser removido (dequeue).
- **Operações Principais:**
  - **enqueue** (inserir): insere um novo elemento ao final da fila.
  - **dequeue** (remover): remove o elemento do início da fila.
  - **peek** ou **front**: obtém (sem remover) o elemento na frente da fila.
  - **isEmpty**: verifica se a fila está vazia.
- **Aplicações:** Ambas podem ser usadas em algoritmos de BFS (busca em largura), gerenciadores de tarefas, sistemas de filas de mensagens etc.

Diferenças entre Filas em Java e Python

- **Estruturas Disponíveis:**
  - **Java:** Interface `Queue` em `java.util` (e.g., `LinkedList`, `ArrayDeque`, `PriorityQueue`).
  - **Python:** Módulo `queue` (`Queue`, `PriorityQueue`), `collections.deque`.
- **Performance:**
  - **Java:** Escolha da implementação impacta enfileiramento e desenfileiramento.
  - **Python:** `deque` é otimizado; `list` não é ideal para remover do início (O(n)).
- **Sintaxe:**
  - **Java:** `Queue<E> fila = new LinkedList<>();` (ou outra implementação).
  - **Python:** `from collections import deque`; `fila = deque()`.

Comparação entre Pilha e Fila

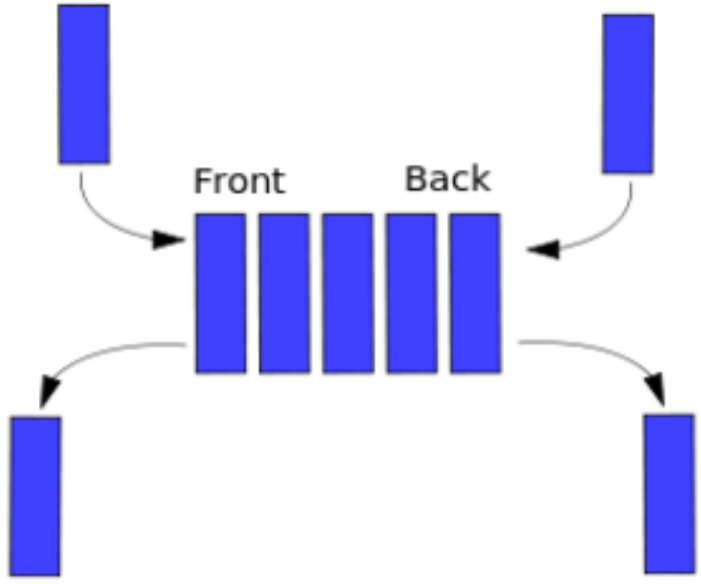
Aspecto	**Pilha**	**Fila**
Princípio	LIFO	FIFO
Operação de Inserção	`push` (no topo)	`enqueue` (no final)
Operação de Remoção	`pop` (do topo)	`dequeue` (do início)
Exemplos	Pilha de pratos	Fila de banco

Deques

Deques

- Deque suporta inserção e remoção tanto na frente quanto no final da fila.
- Deque é pronunciado como "deck"para evitar confusão com o método dequeue das filas regulares.
- O deque é mais geral do que as pilhas e filas, sendo útil em diversas aplicações.

Exemplo: Em um restaurante, uma pessoa pode ser removida da fila e reinserida na primeira posição, ou um cliente no final pode sair devido à impaciência.



O Tipo Abstrato de Dados Deque

Métodos Principais:

- `D.add_first(e)`: Adiciona o elemento e à frente do deque `D`.
- `D.add_last(e)`: Adiciona o elemento e ao final do deque `D`.
- `D.delete_first( )`: Remove e retorna o primeiro elemento do deque `D`; um erro ocorre se o deque estiver vazio.
- `D.delete_last( )`: Remove e retorna o último elemento do deque `D`; um erro ocorre se o deque estiver vazio.

Acessores:

- `D.first( )`: Retorna (sem remover) o primeiro elemento do deque `D`; um erro ocorre se o deque estiver vazio.
- `D.last( )`: Retorna (sem remover) o último elemento do deque `D`; um erro ocorre se o deque estiver vazio.
- `D.is_empty( )`: Retorna `True` se o deque `D` não contiver elementos.
- `len(D)`: Retorna o número de elementos no deque `D`; em Python, isso é implementado com o método especial `len`.

Deque

Operation	Return Value	Deque
D.add_last(5)	–	[5]
D.add_first(3)	–	[3, 5]
D.add_first(7)	–	[7, 3, 5]
D.first( )	7	[7, 3, 5]
D.delete_last( )	5	[7, 3]
len(D)	2	[7, 3]
D.delete_last( )	3	[7]
D.delete_last( )	7	[ ]
D.add_first(6)	–	[6]
D.last( )	6	[6]
D.add_first(8)	–	[8, 6]
D.is_empty( )	False	[8, 6]
D.last( )	6	[8, 6]

Similaridades entre Deques em Java e Python

Conceito de Deque (Double-Ended Queue)

- Em ambas as linguagens, uma deque permite inserção e remoção de elementos tanto na extremidade inicial quanto na extremidade final.
- Oferece a flexibilidade de uso como fila (FIFO), pilha (LIFO) ou uma combinação das duas abordagens.

Operações Principais

- **addFirst / addLast** (inserir no início / fim): Insere elemento na extremidade escolhida.
- **removeFirst / removeLast** (remover do início / fim): Remove e retorna o elemento na extremidade.
- **peekFirst / peekLast**: Retorna (sem remover) o elemento no início ou no fim.
- **isEmpty**: Verifica se a deque está vazia.

Diferenças entre Deques em Java e Python

- **Classes Disponíveis**:
  - **Java**: Interface `Deque` (pacote `java.util`), com implementações como `ArrayDeque` e `LinkedList`.
  - **Python**: A classe `collections.deque` é a implementação principal de deque (nativa do módulo `collections`).
- **Desempenho**:
  - **Java**: `ArrayDeque` normalmente oferece inserções e remoções em O(1) amortizado nas duas pontas; `LinkedList` também implementa `Deque` mas pode ter overhead de ponteiros.
  - **Python**: `collections.deque` é otimizada para inserção e remoção em ambas as extremidades em O(1) amortizado.
- **Sintaxe**:
  - **Java**: `Deque<String> deque = new ArrayDeque<>();` (métodos `addFirst`, `addLast`, `removeFirst`, etc.)
  - **Python**: `from collections import deque` e `d = deque();` (métodos `appendleft`, `append`, `popleft`, `pop`, etc.)

Considerações

Considerações

- **Recordando!!!**
  - Introdução a estruturas lineares.
  - Pilhas (Stack).
  - Filas (Queues).
  - Deques
  - Exercícios.

  Referências:

  - Goodrich, Michael T., Tamassia, Roberto, e Goldwasser, Michael H. *Data Structures and Algorithms in Python*. Wiley, Hoboken, NJ, 2013.
  - Goodrich, Michael T. e Tamassia, Roberto. *Data Structures and Algorithms in Java*. 6ª Edição, Wiley, Hoboken, NJ, 2016.