

Estruturas de Dados com Python

Filas

Aula 26/04/2022

Curso: Bacharelado em Sistemas de Informação

Professor: Claudio Eduardo Oliveira

Roteiro da Aula

- Filas
 - Introdução
 - Filas com Estruturas Encadeadas
 - Filas - inserção
 - Filas - remoção
 - Utilização da implementação de fila
 - Filas em Python (biblioteca collections.deque)

Introdução

Filas são estruturas de dados em que só é possível inserir um novo elemento no final da fila e só é possível remover um elemento do início. Dizemos que filas seguem um protocolo em que o primeiro a entrar é o primeiro a sair. Filas são geralmente implementadas com arranjos.

Em uma fila, para que o primeiro elemento a entrar (ser inserido) seja o primeiro a sair (ser removido), precisamos ser capazes de:

- Inserir um novo elemento no final da fila.
- Remover um elemento do início da fila.

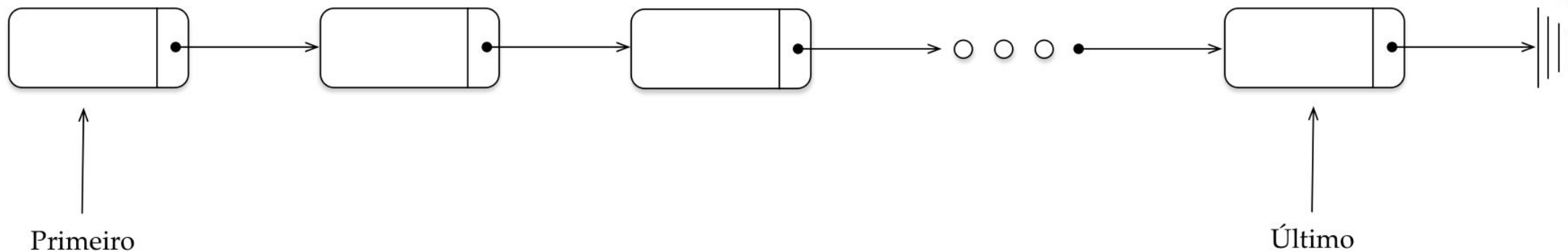
Existe uma forma de usarmos listas como se fossem filas em Python. Mas isso não é muito eficiente, por causa da forma como Python implementa listas. Felizmente, existe uma biblioteca que implementa as operações de filas em Python.

Nesta seção criaremos classes com nossas próprias implementações de filas em Python e nas próximas seções mostraremos como usar a biblioteca deque do Python para implementar uma fila.

Filas com Estruturas Encadeadas

Anteriormente, mostramos como implementar uma pilha usando uma estrutura encadeada. Nesta seção, implementaremos uma fila usando uma estrutura encadeada.

Em filas, inserções ocorrem no final e remoções ocorrem no começo. Para isso, usaremos dois ponteiros: um para o começo da fila, e outro para o final. Esses ponteiros nos permitirão implementar inserções e remoções com custo constante.



Filas com Estruturas Encadeadas

O exemplo abaixo tenta deixar os conceitos mais claros.

```
class Nodo:
    """Esta classe representa um nodo de uma estrutura duplamente encadeada."""
    def __init__(self, dado=0, proximo_nodo=None):
        self.dado = dado
        self.proximo = proximo_nodo

    def __repr__(self):
        return '%s -> %s' % (self.dado, self.proximo)

class Fila:
    """Esta classe representa uma fila usando uma estrutura encadeada."""

    def __init__(self):
        self.primeiro = None
        self.ultimo = None

    def __repr__(self):
        return "[" + str(self.primeiro) + "]"
```

Filas - inserção

A figura mostra a implementação de uma fila com uma estrutura encadeada, semelhante à que usamos no caso das listas e das pilhas. Tomando como base essa implementação, podemos desenvolver funções para inserir e remover elementos da fila.

Na figura abaixo mostramos as etapas da inserção de um novo elemento em uma fila.

Filas - inserção

O código a seguir mostra a implementação de uma função que insere um elemento em uma fila.

```
def insere(self, novo_dado):  
    """Insere um elemento no final da fila."""  
  
    # Cria um novo nodo com o dado a ser armazenado.  
    novo_nodo = Nodo(novo_dado)  
  
    # Insere em uma fila vazia.  
    if self.primeiro == None:  
        self.primeiro = novo_nodo  
        self.ultimo = novo_nodo  
    else:  
        # Faz com que o novo nodo seja o último da fila.  
        self.ultimo.proximo = novo_nodo  
  
        # Faz com que o último da fila referencie o novo nodo.  
        self.ultimo = novo_nodo
```

Filas - remoção

Na figura abaixo mostramos as etapas da remoção de um elemento de uma fila.

Filas - remoção

E o seguinte código mostra a implementação de uma função que remove um elemento de uma fila.

```
def remove(self):  
    """Remove o último elemento da fila."""  
  
    assert self.primeiro != None, "Impossível remover elemento de fila vazia."  
  
    self.primeiro = self.primeiro.proximo  
  
    if self.primeiro == None:  
        self.ultimo = None
```

Novamente, diferentemente do que ocorre com pilhas, inserções e remoções acontecem em extremos opostos da estrutura: inserções ocorrem no final e remoções ocorrem no começo da fila.

Utilização da implementação de fila

Vejamos agora como usar nossa implementação de fila.

```
# Cria uma fila vazia.
fila = Fila()
print("Fila vazia: ", fila)

# Insere elementos na fila.
for i in range(5):
    fila.insere(i)
    print("Insere o valor {0} final da fila: {1}".format(i, fila))

# Remove elementos da fila.
while fila.primeiro != None:
    fila.remove()
    print("Removendo elemento que está no começo da fila: ", fila)
```

Utilização da implementação de fila

Vejamos agora a saída para o código da implementação de fila.

```
Fila vazia: [None]
Insere o valor 0 final da fila: [0 -> None]
Insere o valor 1 final da fila: [0 -> 1 -> None]
Insere o valor 2 final da fila: [0 -> 1 -> 2 -> None]
Insere o valor 3 final da fila: [0 -> 1 -> 2 -> 3 -> None]
Insere o valor 4 final da fila: [0 -> 1 -> 2 -> 3 -> 4 -> None]
Removendo elemento que está no começo da fila: [1 -> 2 -> 3 -> 4 -> None]
Removendo elemento que está no começo da fila: [2 -> 3 -> 4 -> None]
Removendo elemento que está no começo da fila: [3 -> 4 -> None]
Removendo elemento que está no começo da fila: [4 -> None]
Removendo elemento que está no começo da fila: [None]
```

Filas em Python (biblioteca collections.deque)

Nesta seção usaremos a biblioteca `collections.deque` para ilustrarmos as operações de inserção e remoção de elementos em filas. O exemplo ao lado ilustra o funcionamento das operações de inserção e remoção de elementos em uma fila. Inserções acontecem à direita (no final) da fila e remoções ocorrem à esquerda (no começo) da fila. Se implementadas cuidadosamente, essas operações possuem complexidade constante, ou seja, $O(1)$.

```
from collections import deque

# Cria uma fila com três elementos.
fila = deque(["Banana", "Maçã", "Pera"])
print("Fila: ", fila)

# Adiciona um elemento ao final da fila.
fila.append("Uva")
print("Adicionando um elemento: ", fila)

# Remove o primeiro elemento adicionado à fila.
fila.popleft()
print("Removendo um elemento: ", fila)
```

```
Fila: deque(['Banana', 'Maçã', 'Pera'])
Adicionando um elemento: deque(['Banana', 'Maçã', 'Pera', 'Uva'])
Removendo um elemento: deque(['Maçã', 'Pera', 'Uva'])
```

Bibliografia

- <https://algoritmoempython.com.br/cursos/algoritmos-python/estruturas-dados/filas/>