

# MC102 - Algoritmos e Programação de Computadores

## Lista de Exercícios 7

6 de Junho de 2018

1. Suponha que criamos uma classe para representar produtos de um supermercado:

```
class Produto():
    def __init__(self, nome, preco, quant):
        self.nome = nome
        self.preco = preco
        self.quantidade = quant
```

Implemente duas funções que recebem uma lista de produtos como parâmetro e fazem a ordenação da lista. Uma função deixa a lista ordenada por preços e a outra deixa a lista ordenada pela quantidade de itens no estoque. As funções devem implementar algum dos algoritmos de ordenação vistos em sala de aula, devendo-se alterar a maneira como se comparam dois objetos.

Os protótipos são:

```
def ordenaPreco(vetProduto)
def ordenaQuant(vetProduto)
```

2. Suponha que criamos uma classe para representar Datas:

```
class Data():
    def __init__(self, dia, mes, ano):
        self.dia = dia
        self.mes = mes
        self.ano = ano
```

Implemente uma função que recebe uma lista de Datas como parâmetro e que deixe a lista ordenada crescentemente por ordem cronológica. Protótipo da função é:

```
def ordena(vetData)
```

3. Suponha que criamos uma classe para representar pessoas e uma outra classe para representar um cadastro de várias pessoas.

```
class Pessoa():
    def __init__(self, rg, cpf, nome):
        self.rg = rg
        self.cpf = cpf
```

```
        self.nome = nome

class Base():
    def __init__(self):
        self.pessoas = []
```

Crie métodos na classe Base para cada uma das operações abaixo:

- Inclui Pessoa: este método recebe como parâmetro um objeto do tipo Pessoa e o inclui na base caso já não exista na base uma pessoa com o mesmo RG ou CPF. O método deve lançar uma exceção ValueError caso haja uma pessoa cadastrada com o mesmo RG ou CPF. **OBS:** A inclusão de uma pessoa deve manter a lista **pessoas** em ordem crescente por valor de RG de tal forma que se possa realizar a busca binária depois (veja Busca).

```
def inclui(self, pessoa)
```

- Exclui Pessoa: este método recebe como parâmetro um int representando o RG de uma pessoa e o exclui da base caso esteja presente. O método deve lançar uma exceção ValueError caso não haja pessoa com o RG informado.

```
def remove(rg)
```

- Busca: este método recebe como parâmetro um int representado o RG de uma pessoa e realiza uma busca binária na lista para encontrar a pessoa com RG informado. O método deve lançar uma exceção ValueError caso não haja pessoa com o RG informado.

4. Refaça as funções de busca sequencial e busca binária vistas em aula assumindo que o vetor possui chaves que podem aparecer repetidas. Neste caso, você deve retornar em uma lista todas as posições onde a chave foi encontrada.

Protótipo da função: **void busca(vet, chave)**