

Lista 5 – Algoritmos de Pesquisa em JAVA

Para cada problema proposto postar as soluções no Moodle. Para isto, compacte em único arquivo o conjunto das soluções (os arquivos com extensão .JAVA).

Para as questões a seguir vocês devem utilizar o *ArrayList* que é a biblioteca nativa do Java que implementa listas. Dessa forma vocês deverão inicializar uma lista de String na main como descrito a seguir:

- Inclua as bibliotecas:

```
import java.util.List;  
import java.util.ArrayList;
```

- Forma 1

```
ArrayList<String> Lstring= new ArrayList<>();//Cria uma Lista de  
String
```

- Forma 2

```
ArrayList<String> Lstring= new ArrayList<String>();// Cria uma lista  
de string
```

- Comparação de *String* em um *ArrayList*

```
ArrayList<String> frutas = new ArrayList<>(Arrays.asList("banana",  
"laranja", "maçã", "melancia", "uva"));  
  
    // Verifica se a posição 2 contém exatamente "maçã"  
    if (frutas.size() > 2 && frutas.get(2).equals("maçã")) {  
        System.out.println("A posição 2 contém 'maçã'");  
    } else {  
        System.out.println("A posição 2 NÃO contém 'maçã'");  
    }  
}
```

1. Cria um procedimento que receba um *ArraList<String>* como parâmetro e adicione N nomes inseridos pelo usuário até que o usuário digite a FLAG “STOP” para encerrar a inserção. DICA *Lstring.add("Neymar");* // Adiciona o nome Neymar na lista

2. Crie um procedimento que receba como parâmetro um *ArraList<String>* e imprima todos os valores da lista. DICA *Lstring.get(5);*//acessando a posição 5 da lista

3. Cria uma função booleana que receba como parâmetro a lista de nomes e uma *String* e faça o algoritmo de busca sequencial iterativa na Lista e retorne verdadeiro caso encontre o ou falso em caso contrário.

4. Cria uma função booleana que receba como parâmetro a lista de nomes e uma *String* e

faça o algoritmo de busca binária iterativa na Lista e retorne verdadeiro caso encontre-o ou falso em caso contrário. Você deve ordenar a lista com uso da biblioteca nativa JAVA. DICA `import java.util.Collections;`
`Collections.sort(Lstring); // método nativo que ordena em ordem crescente`

5. Cria um procedimento que receba a lista de String e uma string com um nome, e faça uma busca sequencial na lista e remova a primeira ocorrência do nome, se caso este seja encontrado. Dica `Lstring.remove(7); // remove o oitavo elemento da lista`

6. Cria um procedimento que receba a lista de String e uma string com um nome, e faça uma busca binária na lista e remova a primeira ocorrência do nome, se caso este seja encontrado. Dica `Lstring.remove(7); // remove o oitavo elemento da lista`

7. Crie uma versão recursiva para busca sequencial.

8. Crie uma versão recursiva para busca binária.

Informações sobre cópias

As questões são individuais. Em caso de cópias de trabalho a pontuação será zero para os autores originais e copiadores. Não serão aceitas justificativas como: “Fizemos o trabalho juntos, por isso estão idênticos”.

Para descontrair

**Por que minha função
não retorna nada...**



**Ahh, eu não chamei
a função 😓**



