

Lista 11 – Pesquisa Sequencia, Pesquisa Binária, Árvore Binária de Pesquisa

- 1) Faça um programa que use o método de pesquisa binária para encontrar e imprimir a posição de um determinado elemento no vetor. Se o elemento não existir, seu programa deve imprimir o valor -1.

Digite a quantidade de elementos do vetor: 5

Digite os elementos do vetor: 1 2 3 4 5

Digite o elemento procurado: 3

Posição do elemento procurado: 2

Dica- implemente o seguinte método: `int pesquisaBinaria(int[] vet, int n)`

- 2) Adicione o seguinte método na classe `ArvoreBinaria`: `int getMaior()`. O método deve retornar o maior elemento de uma Árvore Binária. (Dica: o maior elemento estará na subárvore direita)
- 3) Adicione o seguinte método na classe `ArvoreBinaria`: `int getMenor()`. O método deve retornar o menor elemento de uma Árvore Binária. (Dica: o menor elemento estará na subárvore esquerda)
- 4) Faça um programa que apresente o seguinte menu de opções:
 - 1- Inserir um número na árvore binária
 - 2- Remover um número da árvore binária
 - 3- Pesquisar um número na árvore binária
 - 4- Mostrar o maior elemento da árvore binária.
 - 5- Mostrar o menor elemento da árvore de pesquisa
 - 6- Mostrar todos os elementos da árvore, usando o caminharmento central
 - 7- Mostrar todos os elementos da árvore, usando o caminharmento pós-ordem.
 - 8- Mostrar todos os elementos da árvore, usando o caminharmento pré-ordem.
 - 9- Sair

O programa deve ler a opção escolhida pelo usuário, ler os dados necessários e fazer o processamento. O programa deve ser executado até que a opção 9 seja escolhida pelo usuário.

- 5) Faça um programa que apresente o seguinte menu de opções:
 - 1- Inserir um nome na árvore binária
 - 2- Remover um nome da árvore binária
 - 3- Pesquisar um nome na árvore binária
 - 4- Mostrar todos os elementos da árvore, usando o caminharmento central
 - 5- Mostrar todos os elementos da árvore, usando o caminharmento pós-ordem.
 - 6- Mostrar todos os elementos da árvore, usando o caminharmento pré-ordem.
 - 7- Sair

O programa deve ler a opção escolhida pelo usuário, ler os dados necessários e fazer o processamento. O programa deve ser executado até que a opção 7 seja escolhida pelo usuário.

Dica: Na classe `Nó` altere o atributo `elemento` de `int` para `String`. Na classe `ArvoreBinaria` faça as alterações necessárias nos métodos para manipular Strings. Para comparar Strings use o método `compareToIgnoreCase()`

[Java String compareToIgnoreCase\(\) Method \(w3schools.com\)](https://www.w3schools.com/java/compare_to_ignore_case.asp)

- 6) Árvore binária de pesquisa:
 - a. Mostre (desenhe) uma árvore binária de pesquisa após a inserção dos seguintes elementos (nessa ordem): 10, 20, 5, 8, 12, 22, 23, 24, 11, 13, 18

- b. Mostre (desenhe) como ficará a árvore acima após a remoção dos seguintes elementos (nessa ordem): 22, 11, 10
- c. Liste as sequências de nós encontrados ao percorrer a árvore do **item A** em pré-ordem, pós-ordem e central

7) Árvore binária de pesquisa:

- a. Mostre (desenhe) uma árvore binária de pesquisa após a inserção dos seguintes elementos (nessa ordem): 6, 5, 8, 4, 7, 12, 2, 10, 15, 1, 3, 9
- b. Mostre (desenhe) como ficará a árvore acima após a remoção dos seguintes elementos (nessa ordem): 9, 4, 12
- c. Liste as sequências de nós encontrados ao percorrer a árvore do **item A** em pré-ordem, pós-ordem e central