

Atividade 01

Criar uma classe que inicializa um vetor e possui os seguintes métodos para preenchimento:

Crescente / Decrescente / Aleatório

Obs: No método aleatório, utilizar a função de setar a semente antes do laço for, para que os resultados dos exercícios sejam os mesmos para todos alunos.

Criar um código que busque sequencialmente o valor 78 em um vetor. Testar com vetores de tamanhos 50, 500 e 5000. Variar o teste com o vetor inicializado de forma crescente, decrescente e aleatório com raiz 3 e números entre 0 e 199.

Repetir os testes para busca binária. (menos aleatório).

Em todos os testes mostrar a quantidade de comparações realizadas e tempo relógio gasto.

Enviar um relatório pdf com o algoritmo e o quadro com os resultados. Fazer uma análise e conclusão dos resultados. Não precisa ser grande

Código

```
import java.util.Random;
import java.util.Scanner;

public class AtividadeBusca {
    Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        AtividadeBusca atividadeBusca = new AtividadeBusca();
        atividadeBusca.criarVetor();
    }

    private void criarVetor() {
        int forma;

        System.out.println("Escolha uma forma para preencher o vetor ");
        System.out.println(" 1 - Crescente \n 2 - Decrescente \n 3 - Aleatório");
        forma = scanner.nextInt();

        if (forma == 1) {
            vetorCrescente();
        } else if (forma == 2) {
            vetorDecrescente();
        } else if (forma == 3) {
            vetorAleatorio();
        } else {
            System.out.println("Opção inválida");
        }
    }

    private void vetorCrescente() {
        System.out.print("Informe o tamanho do vetor a ser criado:");
        int tamanho = scanner.nextInt();

        int vet[] = new int[tamanho];

        for(int i = 0; i < vet.length; i++){
            vet[i] = i;
        }
    }
}
```

```
}

mostrarVetor(vet);
buscaSequencial(vet);
buscaBinaria(vet);

}

private void vetorDecrescente() {

    System.out.print("Informe o tamanho do vetor a ser criado:");
    int tamanho = scanner.nextInt();

    int vet[] = new int[tamanho];

    for(int i = 0; i < vet.length; i++){
        vet[i] = (vet.length - 1) - i;
    }

    mostrarVetor(vet);
    buscaSequencial(vet);
    buscaBinaria(vet);

}

private void vetorAleatorio() {

    Random in = new Random(3);

    System.out.print("Informe o tamanho do vetor a ser criado:");
    int tamanho = scanner.nextInt();

    System.out.println("Informe o intervalo de números para preenchimento do vetor:");
    System.out.print("Inicio: ");
    int inicio = scanner.nextInt();
    System.out.print("Fim: ");
    int fim = scanner.nextInt();

    int vet[] = new int[tamanho];

    for(int i = 0; i < vet.length; i++){
        vet[i] = in.nextInt(inicio, fim);
    }

    mostrarVetor(vet);
    buscaSequencial(vet);

}

private void mostrarVetor(int[] vet) {

    System.out.println("Deseja mostrar o vetor: ");
    System.out.println(" 1 - Sim \n 2 - Não");
    int op = scanner.nextInt();

    if (op == 1 ) {
        for(int i = 0; i < vet.length; i++){
            System.out.print(vet[i]);
            System.out.print(" - ");
        }
    }
}
```

```
    }  
    }  
}  
  
private void buscaSequencial(int[] vet) {  
  
    int cont;  
  
    System.out.println();  
    System.out.println("---Busca sequencial");  
    System.out.println("Informe o número que você deseja buscar no vetor:  
");  
    int num = scanner.nextInt();  
  
    Long hrInicio = System.nanoTime();  
  
    for (cont = 0; cont < vet.length; cont++) {  
        if (vet[cont] == num) {  
            break;  
        }  
    }  
    Long hrFim = System.nanoTime();  
    Double tempoTotal = (hrFim.doubleValue() - hrInicio.doubleValue()) /  
1000000;  
  
    System.out.printf("Tempo gasto na busca: %.4f ", tempoTotal);  
    System.out.println();  
    System.out.println("Comparações realizadas na busca: " + cont);  
  
}  
  
private void buscaBinaria(int[] vet) {  
  
    int meio;  
    int inicio = 0;  
    int fim = vet.length - 1;  
    int comp = 0;  
  
    System.out.println();  
    System.out.println("---Busca binária");  
    System.out.println("Informe o número que você deseja buscar no vetor:  
");  
    int num = scanner.nextInt();  
  
    Long hrInicio = System.nanoTime();  
    while (inicio <= fim) {  
        comp = comp + 1;  
        meio = (inicio + fim) / 2;  
        if (vet[meio] == num) {  
            break;  
        } else if (vet[meio] < num) {  
            inicio = meio + 1;  
        } else if (vet[meio] > num) {  
            fim = meio - 1;  
        } else {  
            break;  
        }  
    }  
    Long hrFim = System.nanoTime();  
    Double tempoTotal = (hrFim.doubleValue() - hrInicio.doubleValue()) /
```

```
1000000;  
  
    System.out.printf("Tempo gasto na busca: %.4f ", tempoTotal);  
    System.out.println();  
    System.out.println("Comparações realizadas na busca: " + comp);  
  
}
```

Resultado

(ms = milissegundos / tempo gasto na execução)

(c = comparações / comparações realizadas na execução)

Vetor	Busca Sequencial			Busca Binária	
	Crescente	Decrescente	Aleatório	Crescente	Decrescente
50	0,0743 ms / 50 c	0,0853 ms / 50 c	0,0766 ms / 50 c	0,0035 ms / 6 c	0,0030 ms / 6 c
500	0,0893 ms / 78 c	0,1082 ms / 421 c	0,0883 ms / 500 c	0,0043 ms / 8 c	0,0029 ms / 8 c
5000	0,0854 ms / 78 c	0,1524 ms / 4921 c	0,0980 ms / 741 c	0,0042 ms / 12 c	0,0038 ms / 12 c

Análise

Podemos considerar que a busca binária é executada utilizando menos comparações e consequentemente em um tempo menor, na busca sequencial o número de comparações pode variar dependendo do tipo do vetor, assim teremos também uma diferença no tempo de execução entre cada tipo de vetor. Definir qual seria a melhor busca, dependeria de uma análise realizada em cada situação a ser implementada.