

Lista 6, 7 – Métodos de ordenação

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

1) Implemente os algoritmos em JAVA: BubbleSort, SelectionSort, InsertSort, Mergesort, QuickSort, ShellSort e HeapSort. Para o tipo **DOUBLE**.

O código a seguir calcula o tempo gasto para executar uma determinada função. Modifique-o para que possa mensurar o tempo estimado para a ordenação dos métodos implementados. Além disso, contém uma função de geração de valores aleatórios e decrescente. DICA: não se esqueça de randomizar o vetor antes de inicializar a contagem do tempo para cada algoritmo.

```
public static void main(String[] args) {

    double [] vet= gerarAleatorio(10000, 1, 50000);

    long startTime = System.nanoTime();

    // Coloque sua função aqui de ordenação
    suaFuncao(...);
    long endTime = System.nanoTime();
    // Calcula o tempo decorrido em milissegundos
    long elapsedTimeInMillis = (endTime - startTime) ;

    System.out.println("Tempo decorrido: " + elapsedTimeInMillis + " nanosegundos");
}

public static void suaFuncao(...) {
    // Implemente sua função aqui
}

// Método para gerar um vetor de valores aleatórios double dentro de um intervalo
public static double[] gerarAleatorio(int tamanho, double minimo, double maximo) {
    double[] vetorAleatorio = new double[tamanho];
    Random random = new Random();

    for (int i = 0; i < tamanho; i++) {
        vetorAleatorio[i] = minimo + (maximo - minimo) * random.nextDouble(); //
        valores dentro do intervalo
    }
}
```

```

        return vetorAleatorio;
    }

// Método para gerar um vetor decrescente com aleatorização da parte fracionada
public static double[] gerarDecrescenteAleatorio(int tamanho) {
    double[] vetorDecrescenteAleatorio = new double[tamanho];
    Random random = new Random();

    for (int i = 0; i < tamanho; i++) {
        double valor = tamanho - i; // gera o valor decrescente

        // Aleatoriza a parte fracionada
        double parteFracionadaAleatoria = random.nextDouble();
        valor += parteFracionadaAleatoria;

        vetorDecrescenteAleatorio[i] = valor;
    }

    return vetorDecrescenteAleatorio;
}

```

2) Execute os algoritmos e anote os resultados na tabela a seguir. obs: execute o algoritmo em uma mesma máquina.

Tabela 1 Vetor já ordenado

Tamanho do Vetor	BubbleSort	SelectionSort	InsertSort	Mergesort	Quicksort	Shellsort	HeapSort
10000							
15000							
30000							

Tabela 2 Vetor Aleatório

Tamanho do Vetor	BubbleSort	SelectionSort	InsertSort	Mergesort	Quicksort	Shellsort	HeapSort
10000							
15000							
30000							

Tabela 3 Vetor Decrescente

Tamanho do Vetor	BubbleSort	SelectionSort	InsertSort	Mergesort	Quicksort	Shellsort	HeapSort
10000							
15000							
30000							

3) Faça uma análise a respeito dos resultados obtidos no seu benchmark das tabelas anteriores.

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”.