

Trabalho Prático

Regras Básicas

1. extends Trabalho Prático 03
2. Fique atento ao Charset dos arquivos de entrada e saída.
3. Nos exercícios de ordenação ou estruturas de dados, se dois objetos tiverem a mesma chave de pesquisa, eles serão ordenados pelo nome da série..



Você foi contratado para trabalhar em uma empresa que distribui *stream* de séries de TV na web. Sua tarefa é organizar as informações das séries disponíveis para exibição ao usuário. Entretanto, esses dados estão espalhados em vários arquivos no formato *html*, os quais foram obtidos através de consultas à base de dados Wikipedia. Todos esses arquivos estão agrupados no arquivo *series.zip*, e o mesmo deve ser descompactado na pasta */tmp/*.¹ Para isso, você deve ler, organizar e armazenar os dados de cada série em memória, utilizando as estruturas de dados em aula (Lista, Pilhas e Filas). Em seguida executar as operações descritas nos arquivos de entrada. Muito cuidado ao realizar o *parser* do texto. Fique atento a descrição dos dados que serão lidos e manipulados pelo seu sistema.

Algoritmos de Ordenação

1. **Ordenação por Seleção em Java:** Na classe *Lista*, implemente o algoritmo de ordenação por seleção considerando que a chave de pesquisa é o atributo **paísDeOrigem**. A entrada e a saída padrão são iguais às da primeira questão do Trabalho Prático II, contudo, a saída corresponde

¹Quando reiniciamos o Linux, ele normalmente apaga os arquivos existentes na pasta */tmp/*.

aos objetos ordenados. Além disso, crie um arquivo de log na pasta corrente com o nome `matricula_selecao.txt` com uma única linha contendo sua matrícula, número de comparações (entre elementos do *array*), número de movimentações (entre elementos do *array*) e o tempo de execução do algoritmo de ordenação. Todas as informações do arquivo de log devem ser separadas por uma tabulação `'\t'`.

2. **Ordenação por Seleção Recursiva em C:** Repita a questão anterior, contudo, usando a Seleção Recursiva. A entrada e a saída padrão serão iguais às da questão anterior. O nome do arquivo de log será `matricula_selecaoRecursiva.txt`.
3. **Ordenação por Inserção em Java:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo de Inserção, fazendo com que a chave de pesquisa seja o atributo **Idioma**. O nome do arquivo de log será `matricula_insercao.txt`.
4. **Shellsort em C:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo Shellsort, fazendo com que a chave de pesquisa seja o atributo **Idioma**. O nome do arquivo de log será `matricula_shellsort.txt`.
5. **Heapsort em Java:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo Heapsort, fazendo com que a chave de pesquisa seja o atributo **Formato**. O nome do arquivo de log será `matricula_heapsort.txt`.
6. **Quicksort em Java:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo Quicksort, fazendo com que a chave de pesquisa seja o atributo **paisDeOrigem**. O nome do arquivo de log será `matricula_quicksort.txt`.
7. **Counting Sort em C:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo Counting Sort, fazendo com que a chave de pesquisa seja o atributo **numeroDeTemporadas**. O nome do arquivo de log será `matricula_countingsort.txt`.
8. **Bolha em Java:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo da Bolha, fazendo com que a chave de pesquisa seja o atributo **numeroDeTemporadas**. O nome do arquivo de log será `matricula_bolha.txt`.
9. **Mergesort em Java:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo Mergesort, fazendo com que a chave de pesquisa seja o atributo **numeroDeEpisodios**. O nome do arquivo de log será `matricula_mergesort.txt`.