



# Trabalho Prático V - Pesquisa e Ordenação

---

## 1 Regras Básicas

1. extends TP3RegrasBasicas;
2. Nos exercícios de ordenação ou estruturas de dados, se dois objetos tiverem a mesma chave de pesquisa, eles serão ordenados pelo ID do objeto.

## 2 Descrição

1. **Pesquisa Sequencial:** Faça a inserção de alguns objetos no final de uma Lista e, em seguida, faça algumas pesquisas sequenciais. A chave primária de pesquisa será o atributo **id**. A entrada padrão é composta por duas partes onde a primeira é igual a entrada da primeira questão do Trabalho Prático III. As demais linhas correspondem a segunda parte. A segunda parte é composta por várias linhas. Cada uma possui um elemento que deve ser pesquisado na Lista. A última linha terá a palavra FIM. A saída padrão será composta por várias linhas contendo as palavras SIM/NÃO para indicar se existe cada um dos elementos pesquisados. Além disso, crie um arquivo de log na pasta corrente com o nome matrícula\_sequencial.txt com uma única linha contendo sua matrícula, tempo de execução do seu algoritmo e número de comparações. Todas as informações do arquivo de log devem ser separadas por uma tabulação '\t'.
2. **Pesquisa Binária:** Repita a questão anterior, contudo, usando a Pesquisa Binária. A entrada e a saída padrão serão iguais às da questão anterior. O nome do arquivo de log será matrícula\_binaria.txt. A entrada desta questão está ordenada.
3. **Ordenação por Seleção:** Na classe Lista, implemente o algoritmo de ordenação por seleção considerando que a chave de pesquisa é o atributo **anoAtualização**. A entrada e a saída padrão são iguais às da primeira questão do Trabalho Prático III, contudo, a saída corresponde aos objetos ordenados. Além disso, crie um arquivo de log na pasta corrente com o nome matrícula\_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'.

4. **Ordenação por Seleção Recursiva:** 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`.
5. **Ordenação por Inserção:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo de Inserção, fazendo com que a chave de pesquisa seja o atributo **nome**. O nome do arquivo de log será `matricula_insercao.txt`.
6. **Shellsort:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo Shellsort, fazendo com que a chave de pesquisa seja o atributo **UF (string)**. O nome do arquivo de log será `matricula_shellsort.txt`.
7. **Heapsort:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo Mergesort, fazendo com que a chave de pesquisa seja o atributo **codigoUF**. O nome do arquivo de log será `matricula_heapsort.txt`.
8. **Quicksort:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo Quicksort, fazendo com que a chave de pesquisa seja o atributo **população**. O nome do arquivo de log será `matricula_quicksort.txt`.
9. **Counting Sort:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo Mergesort, fazendo com que a chave de pesquisa seja o atributo **númeroFuncionários**. O nome do arquivo de log será `matricula_countingsort.txt`.
10. **Bolha:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo da Bolha, fazendo com que a chave de pesquisa seja o atributo **escolaridade**. O nome do arquivo de log será `matricula_bolha.txt`.
11. **Mergesort:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo Mergesort, fazendo com que a chave de pesquisa seja o atributo **númeroComissionados**. O nome do arquivo de log será `matricula_mergesort.txt`.
12. **Radixsort:** Repita a questão de Ordenação por Seleção, contudo, usando o algoritmo Radixsort, fazendo com que a chave de pesquisa seja o valor doidão. O nome do arquivo de log será `matricula_radixsort.txt`. O valor doidão é igual a **atualizaçãoPlano \* 1000 + id**.