

## **Estruturas de Dados II**

### **Vanessa Braganholo**

### **Lista de Exercício 1**

#### **Arquivos Sequenciais**

**ATENÇÃO:** Para todos os exercícios desta lista, assuma que existe uma operação *LeRegistro* e outra *GravaRegistro*, que lê/grava um registro inteiro no/do arquivo. Assim não precisamos nos preocupar com a estrutura do registro nos exercícios.

1) Faça um algoritmo que intercala  $n$  arquivos usando árvore de vencedores.

Parâmetros de entrada do algoritmo: lista de arquivos a serem ordenados,  $n$  (tamanho da lista de arquivos a ordenar)

Parâmetros de saída do algoritmo: arquivo ordenado, chamado *saida.dat*

Assumir que no final de cada arquivo existe um registro com *HIGH\_VALUE* gravado.

2) Explique porque arquivos sequenciais precisam estar ordenados.

3) Os algoritmos de geração de partições classificadas assume que é possível manipular apenas  $x$  registros em memória de uma vez. No algoritmo de Classificação Interna, se o arquivo a ser particionado tem  $n$  registros, quantas partições serão geradas? Quantos registros, no máximo, terá cada partição?

4) Escreva um algoritmo que implementa o método de Geração de Partições por Classificação Interna, assumindo que só podem ser manipulados  $x$  registros em memória de cada vez.

Parâmetros de entrada do algoritmo: arquivo a ser particionado (*entrada.dat*), número de registros do arquivo ( $n$ ), número de registros que podem ser manipulados de cada vez ( $x$ )

Parâmetros de saída do algoritmo: arquivos de partição (*particao1.dat*, *particao2.dat*, ...)

5) Repita o exercício anterior, agora utilizando o método de Seleção com Substituição.

6) Repita o exercício 6, agora utilizando o método de Seleção Natural.

7) Compare os métodos de Seleção com Substituição e Seleção Natural, em relação ao número de operações de entrada e saída necessários para implementar cada método.

8) O que é chave primária?

9) O que é chave secundária? Quais as desvantagens de se utilizar chave secundária em uma aplicação?

10) O que é um registro?