

Segunda lista de exercícios de Pesquisa e Ordenação de Dados 2013/2  
Prof. Marcia Pasin - UFSM

- 1) A partir de duas listas de processos distintos  $I_1$  e  $I_2$ , sendo que  $I_1$  tem  $m$  processos e  $I_2$  tem  $n$  processos, calcular todas as combinações possíveis. Gerar uma árvore binária com as combinações possíveis. Recuperar uma determinada combinação desta árvore. Remover uma determinada combinação desta árvore.
- 2) A partir de duas listas de processos distintos  $I_1$  e  $I_2$ , sendo que  $I_1$  tem  $m$  processos e  $I_2$  tem  $n$  processos, calcular todas as combinações possíveis. Gerar uma trie com as combinações possíveis. Recuperar uma determinada combinação desta árvore. Remover uma determinada combinação desta árvore.
- 3) Dado um grupo de itens de dados, de  $a$  até  $z$ , armazená-los em uma árvore  $B$ , de ordem  $m$ . Cada nodo da árvore deverá ser representado como um arquivo com, no máximo,  $2m$  itens.
- 4) Dado um grupo de itens de dados, de  $a$  até  $z$ , armazená-los em uma árvore  $B^*$ , de ordem  $m$ . Cada folha da árvore deverá ser representada como um arquivo sequencial com, no máximo,  $2m$  itens. Implementar a função para recuperar um item; computar o número de acessos a bloco de disco para recuperar o item.
- 5) Dado um grupo de itens de dados, de  $a$  até  $z$ , armazená-los em uma árvore  $B^+$ , de ordem  $m$ . Cada folha da árvore deverá ser representada como um arquivo sequencial com, no máximo,  $2m$  itens. Implementar a função para recuperar um item; computar o número de acessos a bloco de disco para recuperar o item.
- 6) Dado um grupo de itens de dados, de  $a$  até  $z$ , armazená-los em um arquivo sequencial-indexado. Cada bloco deverá conter, no máximo,  $2m$  itens. Implementar a função para recuperar um item; computar o número de acessos a bloco de disco para recuperar o item.
- 7) Implementar o algoritmo de compressão de dados Huffman, dado um arquivo de entrada. Implementar também a função inversa (descompressão). Calcular a taxa de compressão do arquivo de entrada.
- 8) Implementar um esquema de índice invertido. A partir de um texto, armazenado em um arquivo, ler a lista de palavras, ordenar se necessário, e adicioná-la em uma estrutura de dados (de sua escolha) para compor o índice invertido. Implementar a operação de busca de uma palavra no índice que deverá retornar sucesso ou "palavra não encontrada".
- 9) Com o uso da API OSM ou da API GoogleMaps, usando uma estrutura de dados multidimensional, implementar a consulta do tipo onde estou e consulta do vizinho mais próximo.
- 10) Com o uso da API OSM ou da API GoogleMaps, usando uma estrutura de dados multidimensional, implementar a consulta de correspondência parcial e consulta em intervalos.