

Exercícios Diversos: Ordenação e Pesquisa

- Para alguns dos algoritmos de pesquisa vistos em sala (Seqüencial, Binária, Árvore Binária de Pesquisa, e Hash):
 - Forneça uma descrição sucinta, em português, do funcionamento do algoritmo.
 - Escreva qual é ordem de complexidade do número de comparações no pior caso, melhor caso e caso médio.
 - Discuta a eficiência da estrutura para as operações de inserção e impressão de todos os registros em ordem.
- Considere a seqüência de chaves Q U E S T A O F C I L e a codificação $A = 0$, $B = 1$, $C = 2$, etc.
 - Desenhe o conteúdo da tabela hash resultante da inserção dos registros com essas chaves nessa ordem em uma tabela inicialmente vazia de tamanho 7 usando listas encadeadas. Use a função hash $h(k) = k \bmod 7$ onde k é a codificação da letra
 - Desenhe o conteúdo da tabela hash resultante da inserção dos registros com essas chaves nessa ordem em uma tabela inicialmente vazia de tamanho 13 usando endereçamento aberto e hash linear para tratar colisões. Use a função hash $h(k) = k \bmod 13$ onde k é a codificação da letra.
- Considere a seqüência de chaves Q U E S T A O F C I L . Mostre passo-a-passo com seria a inserção dessas chaves em uma árvore binária de pesquisa não balanceada.
- Escreva um procedimento que, dada uma árvore binária de pesquisa, determine a sua altura.
- Descreva sucintamente o funcionamento de uma Trie binária. Qual é um de seus principais problemas? Como a árvore Patrícia resolve esse problema?
- Considere a implementação de uma tabela Hash de tamanho $M=11$, com endereçamento aberto utilizando a função $k \bmod M$. Responda as seguintes questões:
 - Mostre a configuração da tabela após a inserção dos registros com as chaves: 4, 17, 13, 35, 25, 11, 2, 10, 32.
 - Mostre a configuração da tabela após a **remoção** dos registros com as chaves: 25, 11.
 - Mostre a configuração da tabela após a inserção dos registros com as chaves: 40, 3.
- Considere a árvore para pesquisa digital vista em sala (Trie).
 - Declare a estrutura de dados necessária para implementá-la.
 - Implemente um procedimento para fazer a pesquisa de uma chave c em um Trie. Considere que c é um inteiro com 8 bits e que você tem uma função $\text{Bit}(c:\text{integer}, i:\text{integer})$ que retorna o i -ésimo bit de c (da esquerda para direita).
- Para o vetor composto pelas letras H E A P S O R T, mostre o resultado da construção do *heap* e configuração do *heap* após cada uma das 3 primeiras execuções do algoritmo Heapsort.
- Qual dos métodos de ordenação estudados em sala é mais adequado para cada uma das aplicações descritas abaixo? **Justifique** cada uma de suas respostas.
 - Você deseja ordenar um conjunto de 200 registros uma única vez.
 - Você quer colocar novos alunos em uma lista de chamada já ordenada.
 - Você deseja escrever um programa que vai ser utilizado milhares de vezes para ordenar um grande número de dados aleatórios.

10. Considere a execução do algoritmo **Quicksort** visto em sala (pivô escolhido no meio do vetor) com o vetor [2 10 1 8 20 5 14 13 9]. Quantas chamadas do procedimento `Partição` serão feitas para ordenar esse vetor? Qual o pivô utilizado em cada uma das chamadas? Quais são as partições (sub-vetores) resultantes de cada uma dessas chamadas?

11. Considere uma grande empresa com 8000 funcionários, onde a maioria dos funcionários possui um ramal de telefônico de 4 dígitos. Ex. 4735 – Luiz, 6833 – Marcos, etc.

- Proponha uma estrutura de dados que seja eficiente tanto para inserção de novos ramais quanto para fazer uma pesquisa pelo funcionário usando o ramal como chave.
- Qual a ordem de complexidade da operação de pesquisa no melhor caso e no pior caso? Explique.
- Considere agora essa empresa foi comprada por outra que também possui cerca de 9000 funcionários e ramais de 4 dígitos. O que deve ser feito para integrar os dois sistemas telefônicos?

12. Dado o vetor abaixo, execute o Quicksort usando como pivô a mediana de 3 (considere o 1º., o último e o elemento do meio). Para subvetores com 5 ou menos elementos execute o método da inserção. Mostre a ordenação passo a passo.

10	1	5	19	12	10	2	20	6	11	3	8	13	7	14	16	4	9	15	17
----	---	---	----	----	----	---	----	---	----	---	---	----	---	----	----	---	---	----	----

13. Dado o vetor abaixo:

1	6	7	2	4	5	8	3
---	---	---	---	---	---	---	---

- O vetor é um heap? Justifique
- Caso não seja, mostre os passos (usando o algoritmo visto em sala) para transformá-lo em heap.
- Mostre os passos necessários para colocar os 2 maiores elementos em suas posições usando o heapsort, e certifique-se de que ao final o heap está válido.