

Aula 04 - Lista de exercícios: Implementação de Buscas

1. Implemente as seguintes funções de busca:
 - a) busca sequencial iterativa sobre um arranjo de inteiros;
 - b) busca sequencial recursiva sobre um arranjo de inteiros;
 - c) busca binária iterativa sobre um arranjo de inteiros;
 - d) busca binária recursiva sobre um arranjo de inteiros.
2. (mostrar) Dada a sequência $A = \{4, 7, 10, 18, 23, 66\}$, realize o rastreamento (em papel) de uma das funções implementadas anteriormente. Considere uma busca pelo elemento 23, e uma busca pelo elemento 19.
3. Usando o gdb, confira se sua implementação bate com o rastreamento realizado.
4. (mostrar) Suponha que $v[0..n-1]$ é um vetor de strings em ordem lexicográfica. Escreva uma função que receba uma string x e devolva um índice m tal que a string x é igual à string $v[m]$. Se tal m não existe, a função deve devolver -1.
5. (mostrar) Suponha que cada elemento do vetor $v[0..n-1]$ é uma struct com dois campos: o nome de um aluno e o número do aluno. Suponha que o vetor está em ordem crescente de números. Escreva uma função de busca binária que receba o número de um aluno e devolva o seu nome. Se o número não está no vetor, a função deve devolver a string vazia. Em um arquivo main.cpp, crie uma função principal para demonstrar o funcionamento da sua função.
6. Suponha uma lista ordenada contendo n itens e um item x que **não** está presente na lista. O problema consiste em determinar entre qual par de itens na lista está o item x , isto é, encontrar $a[i]$ e $a[i+1]$ de tal forma que $a[i] < x < a[i+1]$, para $0 \leq i < n-1$, ou que $x < a[0]$ ou que $x > a[n-1]$.
 - a. Implemente duas funções para resolver esse problema. Cada função deve adotar uma estratégia diferente.
 - b. Teste suas funções com diversos exemplos, e identifique qual das duas é a mais adequada para resolver esse problema.
7. **(Desafio)** Escreva uma função recursiva `busca_ternaria` com o seguinte cabeçalho:

```
int busca_ternaria (int v, int ini, int fim, int x);
```

Sua função deve devolver 1 se x aparece no vetor, 0 em caso contrário. Inspire-se na busca binária. Na busca ternária, você deve dividir o vetor em três (em vez de em dois), comparar o x com os dois elementos separadores dessas três partes e, ou encontra x , ou decide em qual das partes ele pode estar. Procura recursivamente nesta parte.
8. **(Desafio)** Perdido em uma terra muito distante, você se encontra em frente a um muro de comprimento infinito para os dois lados (esquerda e direita). Em meio a uma escuridão total, você carrega um lampião que lhe possibilita ver apenas a porção do muro que se encontra

exatamente à sua frente (o campo de visão que o lampião lhe proporciona equivale exatamente ao tamanho de um passo seu). Existe uma porta no muro que você deseja atravessar. Supondo que a mesma esteja a n passos de sua posição inicial (não se sabe se à direita ou à esquerda), elabore um algoritmo para caminhar ao longo do muro e encontrar a porta em $O(n)$ passos. Considere que n é um valor desconhecido. Considere que a ação composta por dar um passo e verificar a posição do muro correspondente ao custo $O(1)$. Implemente e teste sua solução utilizando um vetor de inteiros.

Sugestão: analise o número de passos do algoritmo que anda 1 passo para esquerda, depois 2 passos para direita, depois 4 para esquerda, depois 8 para direita e assim por diante até encontrar a porta.