# MAPA – Material de Avaliação Prática da Aprendizagem

| | |
|---|---|
| **Acadêmico:** André Luis de Souza Lima | **R.A.:** 21150930-5 |
| **Curso:** Engenharia de Software | |
| **Disciplina:** MAPA - ESOFT - ESTRUTURA DE DADOS II - 52/2023 | |
| **Valor da atividade:** 3,00 | **Prazo:** 08/05/2023 08:00 a 07/07/2023 23:59 |

## Instruções para Realização da Atividade

1. Todos os campos acima deverão ser devidamente preenchidos;
2. É obrigatória a utilização deste formulário para a realização do MAPA;
3. Esta é uma atividade INDIVIDUAL. Caso identificado cópia de colegas, o trabalho de ambos sofrerá decréscimo de nota;
4. Utilizando este formulário, realize sua atividade, salve em seu computador, renomeie e envie em forma de anexo;
5. Formatação exigida para esta atividade: documento Word, Fonte Arial ou Times New Roman tamanho 12, Espaçamento entre linhas 1,5, texto justificado;
6. Ao utilizar quaisquer materiais de pesquisa referencie conforme as normas da ABNT;
7. Critérios de avaliação: Utilização do Template; Atendimento ao Tema; Constituição dos argumentos e organização das Ideias; Correção Gramatical e atendimento às normas ABNT;
8. Procure argumentar de forma clara e objetiva, de acordo com o conteúdo da disciplina.

**Em caso de dúvidas, entre em contato com seu Professor Mediador.**

**Bons estudos!**

## RESPOSTAS

**a)** Caso a chave de busca seja um valor que está ausente dentro do arranjo, qual é o valor que a função buscaBinaria() retornará?

**Resposta:** O retorno da função será igual ao valor -1, indicando que a chave não foi encontrada.

**b)** Para que essa busca funcione, o arranjo precisa, necessariamente, estar ordenado? Em qualquer caso, positivo ou negativo, explique o motivo.

**Resposta:** Não. A busca funcionaria também caso o vetor estivesse desordenado, no entanto, causaria um procedimento ineficiente ao "estartar" o método de busca binária. Uma vez que a busca comece pelo meio do vetor (int meio = i + (f - i)/2;) e dado que a chave de busca fosse menor que o valor do meio do vetor, a busca alternaria para a metade inferior esquerda do vetor ao ser feita uma chamada de retorno com função (return buscaBinaria(arranjo, i, meio-1, chave) como parâmetro. Nesse caso, a função percorreria a metade inferior do vetor. Nesse ponto da iteração, o valor final do arranjo seria a posição do meio menos uma e a inicial a mesma, porém não sendo encontrada a chave na "metade da metade esquerda" do arranjo, e esse *arranjo não estivesse ordenado*, a busca iria, a cada metade da metade, cessando a possibilidade de encontrar a chave, ao passo que a busca poderia ser finalizada, sem encontrar a chave correspondente, que poderia estar na metade superior do arranjo. Então se faz necessário ordenar o arranjo de forma crescente, no intuito de que a busca funcione para o que foi projetada.

**c)** Para que essa busca seja rápida, é preciso aplicar ela em um arranjo estático? Em qualquer caso, positivo ou negativo, explique o motivo.

**Resposta:** Não necessariamente, depende. A busca rápida e eficiente também pode ocorrer em um arranjo dinâmico. O que se discute em relação a busca rápida e eficiente é o tema memória dinâmica e otimização de espaço reservado na memória secundária. Percorrer um espaço em memória reservado estaticamente, e não havendo informação salva "lá" [no "todo" de um vetor estático], é infrutífero em termos de eficiência do programa. Ademais, também é possível que

um volume de dados alocado de forma estática em vetores possua o mesmo volume de que uma alocada dinamicamente. A ideia não é associar busca rápida e eficiente em vetores estáticos somente, mas sim associá-la em vetores ordenados que são alocados na medida em que o programa solicita espaço em memória (alocação dinâmica). Nesse sentido, a busca em uma árvore binária ordenada faz sentido, e faria como se “podasse” subárvores desnecessárias (metades de arranjos) a partir de uma determinada raiz ou nó, atingindo o valor/dado com uma menor iteração possível. Assim, ocupar-se-ia menos processamento e não se criaria “gargalos” na CPU de uma máquina, quando por exemplo o vetor não estiver ordenado ou não organizado em forma de árvore binária. Assim, a busca perfaz um menor caminho, criando atalhos até encontrar um dado, contribuindo para uma busca otimizada se aplicada em memória alocada dinamicamente com certeza.

**d)** Imagine que essa função precisa ser invocada dentro da função main() de um programa em C. Dessa forma, escreva a linha de código (apenas uma linha) que invocaria essa função para realizar a busca em um arranjo denominado VET, que possui 10 elementos e que a chave de busca é igual a 15.

**Resposta:** buscaBinaria(vet, 0, 9, 15);