



Universidade São Judas Tadeu

Ciências da Computação

TESTE DE CAIXA BRANCA – EXERCICIO 01 AULA 04

Victor Ignacio - 823125249

Kaue Brito Vieira - 824211851

Giulia Gabriella - 823122979

Kauê Dib de Souza Dias - 823149871

Vinícius Santana Teixeira - 82319112

Murilo Bonuccelli de Oliveira – 823148988

1. INTRODUÇÃO

O teste de software é uma etapa essencial no processo de desenvolvimento de sistemas, garantindo a verificação e validação do funcionamento correto dos algoritmos. Dentre as diversas abordagens existentes, o teste de caixa branca se destaca por analisar a lógica interna do código, avaliando todos os caminhos possíveis de execução. Este trabalho tem como objetivo aplicar a técnica de teste de caixa branca sobre um algoritmo de busca binária, verificando sua robustez por meio de diferentes cenários de entrada.

A busca binária é um algoritmo eficiente para localizar elementos em um vetor ordenado, realizando divisões sucessivas do espaço de busca. Dada sua ampla utilização, torna-se relevante verificar seu correto funcionamento a partir da análise de seus fluxos de controle, decisões e laços.

2. CÓDIGO ANALISADO

```
public static int busca_binaria(int iVet[], int iK)
{
    int iBaixo, iAlto, iMeio;

    iBaixo=0;
    iAlto=iVet.length-1;
    while(iBaixo <= iAlto)
    {
        iMeio=(iBaixo+iAlto)/2;
        if(iK < iVet[iMeio]) iAlto=iMeio-1;
        else if(iK > iVet[iMeio]) iBaixo=iMeio+1;
        else return iMeio;
    }
    return -1;
}
```

3. METODOLOGIA DO TESTE

Foi aplicada a técnica de cobertura de decisões com o objetivo de percorrer todos os ramos lógicos presentes no algoritmo. Foram analisadas as seguintes estruturas de decisão:

- Condicional *if* para verificar se o valor procurado é menor que o elemento do meio;
- Condicional *else if* para verificar se o valor é maior;
- Condicional *else* para quando o valor é encontrado;
- Estrutura de repetição *while* para percorrer o vetor enquanto houver possibilidade de encontrar o valor.

4. CASOS DE TESTE

A tabela a seguir apresenta os casos de teste elaborados para garantir a cobertura total das estruturas condicionais do código:

Caso	Vetor	iK	Resultado esperado	Caminho percorrido
01	{1,3,5,7,9}	5	2	Valor encontrado na primeira tentativa
02	{1,3,5,7,9}	3	1	Valor menor → entra em iK<
03	{1,3,5,7,9}	9	4	Valor maior → entra em iK>
04	{1,3,5,7,9}	4	-1	Valor inexistente → percorre o laço
05	{10}	10	0	Vetor com um único elemento
06	{}	10	-1	Vetor vazio → laço não executado

5. ANÁLISE DOS RESULTADOS

Todos os testes foram executados de forma lógica, por meio da análise do fluxo de execução do código. Observou-se que o algoritmo cobre adequadamente todas as possibilidades de busca, incluindo:

- Localização do elemento logo na primeira iteração;
- Ajuste de limites quando o valor está à esquerda ou à direita do meio;
- Retorno negativo ao final da execução quando o elemento não está presente.

Além disso, foram considerados cenários extremos, como vetor vazio e vetor com apenas um elemento, garantindo maior robustez à análise.

6. CONCLUSÃO

Através da aplicação do teste de caixa branca ao algoritmo de busca binária, foi possível verificar a correta execução de seus fluxos internos. Os casos de teste elaborados permitiram cobrir todas as estruturas condicionais presentes no código, demonstrando que o algoritmo atende corretamente às expectativas quanto à busca de elementos em vetores ordenados. Dessa forma, conclui-se que o algoritmo analisado é funcional e confiável dentro dos limites testados.