

Estruturas de dados

Busca

Marco A L Barbosa

malbarbo.pro.br

Introdução

- 1) Quais são as operações básicas do TAD Dicionário?
- 2) Como funciona a busca binária?
- 3) O que acontece se uma busca binária for executada em um arranjo não ordenado?
- 4) Por que o tempo de execução da busca binária é $O(\lg(n))$?

Começando

- 5) Implemente o TAD Dicionário usando encadeamento simples.
- 6) Considere um arranjo com os valores 3, 6, 8, 9, 20, 21, 22, 23, 30, 40, 45, 46, 50, 60 e diga quais são os índices dos valores que são comparados com a chave em uma busca binária para cada uma das chaves: 6, 30, 41, 50, 70.
- 7) Modifique a implementação da busca binária para funcionar com arranjos em ordem não crescente.
- 8) Implemente o TAD Dicionário usando arranjo ordenado e busca binária.

Praticando

- 9) Implemente a busca binária usando recursividade.
- 10) Considerando um arranjo ordenado de inteiros onde podem existir valores repetidos, implemente uma busca binária que encontre a primeira ocorrência de um dado número inteiro no arranjo.

Avançando

- 11) Use a ideia da busca binária para implementar um algoritmo que encontre a raiz quadrada inteira, se existir, de um número inteiro positivo.
- 12) Uma invariante de laço é uma propriedade que sempre é verdadeira antes de cada iteração do laço de repetição. A seguir estão alguns exemplos de invariantes

```
def soma(lst: list[int]) -> int:
    pares = 0
    # Invariante
    # pares é a quantidade de valores pares na sublista lst[:i]
    for i in range(len(lst)):
        if lst[i] % 2 == 0:
            pares = pares + 1
    return pares

def cotel(lst: list[int], v: int) -> Bool:
    # Invariante
    # v não está na sublista lst[:i]
    for i in range(len(lst)):
        if lst[i] == v:
```

```
        return True
    return False
```

Enuncie uma invariante para a função de busca binária vista em sala.

Desafios

- 13) Implemente a busca binária de forma que haja apenas uma comparação (dentro do laço) entre a chave e o valor do meio do arranjo.