busca binária

A busca binária é um algoritmo eficiente para encontrar um elemento em uma lista ordenada. O algoritmo funciona dividindo repetidamente o intervalo de busca pela metade até encontrar o elemento desejado ou concluir que ele não está presente.

Aqui está uma implementação em Python de uma função de busca binária:

python

Copiar código

def busca\_binaria(lista, item):

inicio = 0

fim = len(lista) - 1

while inicio <= fim:

meio = (inicio + fim) // 2

chute = lista[meio]

if chute == item:

return meio # Retorna o índice do item encontrado

if chute < item:

inicio = meio + 1 # Ignora a metade esquerda

else:

fim = meio - 1 # Ignora a metade direita

return -1 # Retorna -1 se o item não for encontrado

# Exemplo de uso

numeros = [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]

item\_a\_encontrar = 7

resultado = busca\_binaria(numeros, item\_a\_encontrar)

if resultado != -1:

print(f"Item encontrado na posição {resultado}")

else:

print("Item não encontrado")

**Explicação:**

1. **Função busca\_binaria(lista, item)**: Esta função procura item em uma lista ordenada lista.
   * **inicio e fim**: Inicializam os limites da parte da lista onde a busca é realizada.
   * **Loop while**: Continua enquanto inicio for menor ou igual a fim, indicando que ainda há elementos para verificar.
   * **meio**: Calcula o índice do elemento do meio da lista atual.
   * **chute**: O valor do elemento no índice meio.
   * Se chute é igual a item, a função retorna o índice meio (elemento encontrado).
   * Se chute é menor que item, a busca continua na metade direita da lista (ajustando inicio).
   * Se chute é maior que item, a busca continua na metade esquerda da lista (ajustando fim).
   * Se o loop termina sem encontrar o item, a função retorna -1, indicando que o item não está presente na lista.
2. **Exemplo de uso**: A função é chamada com uma lista de números ordenados e um número a ser encontrado. O resultado indica a posição do item ou que o item não foi encontrado.