



**Data Science
Academy**

www.datascienceacademy.com.br

Introdução à Lógica de Programação

Vamos iniciar a segunda semana do curso gratuito de Introdução à Lógica de Programação. Nesta semana vamos estudar alguns importantes algoritmos, que servem de base para diversos outros algoritmos, inclusive em Machine Learning.

Aula 6 – Pesquisa Binária

Vamos supor que você esteja estudando inglês e queira consultar o dicionário em busca da tradução da palavra oath (que significa juramento).

Como você inicia sua busca no dicionário? Da primeira página? Acho pouco provável. Sabendo a ordem das letras do alfabeto, você provavelmente vai iniciar sua busca a partir da metade do dicionário, afinal não faz sentido pesquisar por todas as páginas até chegar a letra o, certo?

Agora, imagine que você entre no Facebook. Quando faz isso, o Facebook precisa verificar que você tem uma conta no site. Logo, ele procura seu nome de usuário em um banco de dados. Digamos que seu usuário seja kevin. O Facebook poderia começar pelos As e procurar seu nome – mas faz mais sentido que ele comece a busca pelo meio.

Os cenários acima são exemplos de problemas de busca. E todos esses casos usam um algoritmo para resolvê-lo: pesquisa binária. A pesquisa binária é um algoritmo. Sua entrada é uma lista ordenada de elementos. Se o elemento que você está buscando está na lista, a pesquisa binária retorna a sua localização. Caso contrário, a pesquisa binária retorna None.

A pesquisa ou busca binária (em inglês binary search algorithm) é um algoritmo de busca em vetores que segue o paradigma de divisão e conquista.

Elas parte do pressuposto de que o vetor está ordenado e realiza sucessivas divisões do espaço de busca comparando o elemento buscado (chave) com o elemento no meio do vetor. Se o elemento do meio do vetor for a chave, a busca termina com sucesso. Caso contrário, se o elemento do meio vier antes do elemento buscado, então a busca continua na metade posterior do vetor. E finalmente, se o elemento do meio vier depois da chave, a busca continua na metade anterior do vetor.

Este seria o pseudocódigo para a busca binária, usando recursividade:

```
BUSCA-BINÁRIA (V[], início, fim, e)
--i recebe o índice do meio entre início e fim
----se (v[i] = e) entao
-----devolva o índice i # elemento e encontrado
----fimse
----se (início = fim) entao
-----não encontrou o elemento procurado
----senão
-----se (V[i] vem antes de e) então
-----faça a BUSCA-BINÁRIA(V, i+1, fim, e)
-----senão
-----faça a BUSCA-BINÁRIA(V, início, i-1, e)
-----fimse
----fimse
```

Suponha que desejemos procurar pelo elemento 72 em um vetor com os seguintes elementos: [2,6,35,49,56,72,94].

Dado que não sabemos onde o elemento pode estar, nosso palpite é que ele estará no meio do vetor (49, no nosso caso). Nosso palpite está errado, mas ainda assim obtemos uma informação importante: como sabemos que o vetor está ordenado e que o elemento procurado é maior do que o elemento que está no meio do vetor, o elemento procurado tem que estar na segunda metade do vetor, ou seja, entre o sub-vetor [56,72,94].

A seguir, fazemos um novo palpite de onde o elemento pode estar. Como fizemos anteriormente, nosso palpite é que o elemento procurado está no meio do vetor que estamos considerando (segunda metade do vetor). Dessa vez, nosso palpite está certo: o elemento 72 está no meio da segunda metade do vetor. Pesquisa concluída.

A pesquisa binária apresenta complexidade assintótica inferior à pesquisa linear, permitindo-nos pesquisar por elementos com complexidade logarítmica no tamanho do vetor, desde que ele esteja ordenado.

A pesquisa binária também é útil em outros cenários. Por exemplo, suponha que nos seja fornecido um vetor que contém somente os valores 0 e 1 e todos os 0s vêm antes dos 1s. Se desejarmos saber a posição em que o primeiro 1 ocorre, podemos usar pesquisa binária. O raciocínio é o mesmo e o algoritmo é bem parecido, exceto por algumas modificações menores. A pesquisa binária é bastante usada em Visão Computacional, uma das principais áreas da Inteligência Artificial.

Vamos a um exemplo em Python? Para uma clara compreensão, recomendo você executar o código no Jupyter Notebook, SEMPRE atento à indentação.

Programa Python para pesquisa binária recursiva.

Retorna o índice de x no vetor se x estiver presente, caso contrário -1

def pesquisaBinaria (vetor, primeira_pos, ultima_pos, x):

--# Verifica se a última posição do vetor é maior ou igual a 1 para garantir

--# que tenhamos um vetor de comprimento maior que zero

--if ultima_pos >= primeira_pos:

----meio = primeira_pos + (ultima_pos - primeira_pos) // 2

----# Se o elemento estiver presente no meio em si

----if vetor[meio] == x:

-----return meio

----# Se o elemento for menor que o meio, ele poderá estar presente apenas no sub-vetor esquerdo

----elif vetor[meio] > x:

-----return pesquisaBinaria(vetor, primeira_pos, meio-1, x)

----# Senão, o elemento pode estar presente apenas no sub-vetor direito

----else:

-----return pesquisaBinaria(vetor, meio + 1, ultima_pos, x)

--else:

----# O elemento não está presente na matriz

----return -1

Vetor de teste

listaNum = [12, 13, 40, 56, 93]

x = 56

Chamada da função

resultado = pesquisaBinaria(listaNum, 0, len(listaNum)-1, x)

if result != -1:

--print ("O elemento está presente no índice %d!" % resultado)

else:

--print ("O elemento não está presente no vetor!")

Observe que começamos dividindo o vetor ao meio e então fazemos as primeiras checagens. De acordo com o resultado, aplicamos recursividade e chamamos a mesma função, mas agora passando parâmetros diferentes. E ficamos no loop até o resultado final. Observe ainda que para o exemplo dado o resultado é índice 3, pois você já está cansado de saber que a indexação em Python começa por zero!!!

Quer mesmo aprender como aplicar Pesquisa Binária? Então você terá que arregalar as mangas. Resolva o exercício abaixo usando como base o exemplo que forneci a você.

Exercício: Dada a lista de famosos autores brasileiros, crie um programa que pesquise se "Clarice Lispector" está na lista e se estiver indique o índice da posição na lista. Aqui está a lista:

```
autores = ['Monteiro Lobato', 'José de Alencar', 'Cecília Meireles', 'Carlos Drummond de  
Andrade', 'Machado de Assis', 'Clarice Lispector', 'Graciliano Ramos', 'Guimarães Rosa', 'Ruth  
Rocha', 'Luis Fernando Veríssimo']
```

Dica: a lista NÃO está ordenada e a Pesquisa Binária tem como premissa que a lista esteja ordenada. O que fazer? Ordene a lista antes de aplicar a Pesquisa Binária. Você não achou que eu ia te dar moleza, não é?

Dedique um tempo e tente resolver. Somente aplicando o que aprende é que você consolida o conhecimento! Por favor, não poste a solução aqui, pois vou apresentar a solução na Aula 7.

Mãos à obra!

#aula06