



{ Algoritmos E ESTRUTURA DE DADOS }



Busca Por Interpolação



Sumario:

1 - Requisitos.....	3
2 - Conhecendo o algoritmo.....	4
3 - Como funcionar.....	5
4 - Porque e onde usar	7
5 - Complexidade de tempo e espaço.....	8
6 - Vantagens e desvantagens	10

REQUISITOS DO PROJETO:

- A cada execução do programa, ele deve carregar os dados(armazenados em um arquivo texto).
- O programa deve perguntar ao usuário qual cliente ele deseja buscar por nome ou código com o algoritmo Busca por interpolação.
- Compute o tempo de execução do processo de busca.
- Informe a complexidade do algoritmo Busca por Interpolação.





Conhecendo o algoritmo...

O algoritmo de busca por interpolação foi criado por William Wesley Peterson em 1957, é uma técnica de busca em estruturas de dados ordenadas que usa uma fórmula de interpolação para estimar a posição do elemento desejado de forma mais precisa do que a busca binária tradicional.

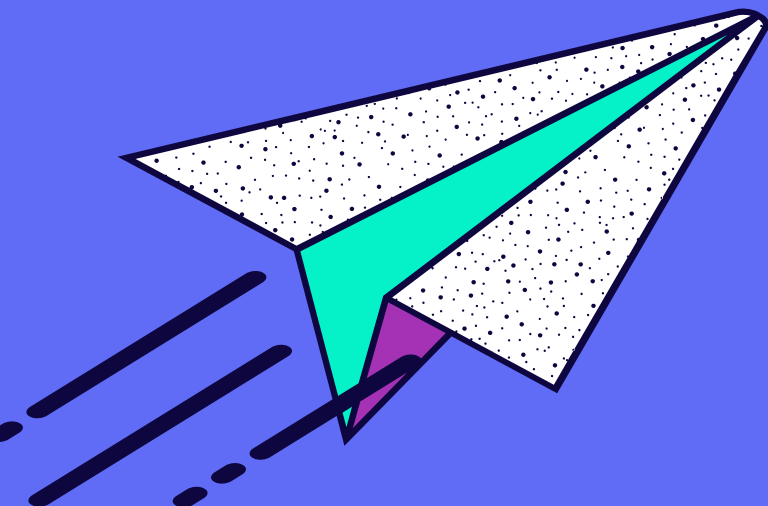


```
1 def busca_interpolacao_codigos(lista_codigos, chave, contagem=False):
2     inicio = 0
3     fim = len(lista_codigos) - 1
4
5     while inicio <= fim and lista_codigos[inicio] <= chave <= lista_codigos[fim]:
6         posicao = inicio + int(((chave - lista_codigos[inicio]) / (
7             lista_codigos[fim] - lista_codigos[inicio])) * (fim - inicio))
8
9         if lista_codigos[posicao] == chave:
10             return posicao # Chave encontrada
11         elif lista_codigos[posicao] < chave:
12             inicio = posicao + 1
13         else:
14             fim = posicao - 1
15     return -1
```



Como funciona?

O algoritmo de busca por interpolação estima a posição do elemento desejado usando uma fórmula e ajusta essa estimativa com base nos valores dos elementos até encontrar o elemento desejado ou determinar sua ausência.



exemplo:

$$m = x - v[c] / v[f] - v[c] * (f - c) + c$$

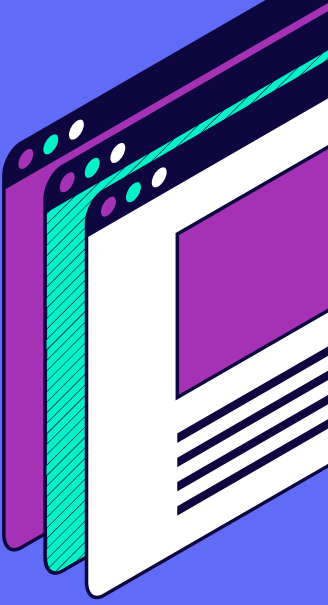
$$m = 2 - 1 / 10 - 1 * (9 - 0) + 0$$

$$m = 1$$

0	1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9	10

0	1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9	10

Demonstração:



Interpolation Search

gbat.com

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	3	7	10	14	15	16	18	20	21	22	23	25	33	35	42	45	47	50	52
$l=0$																			$h=19$

Initialize the array

Porque e onde usar...

em geral a busca por interpolação tem as mesmas aplicações que a binária, no entanto consegue ser mais rápida.

Devemos levar em consideração que a utilização desse algoritmo só faz sentido se formos busca informações ordenadas, como dicionários, agendas e listas.



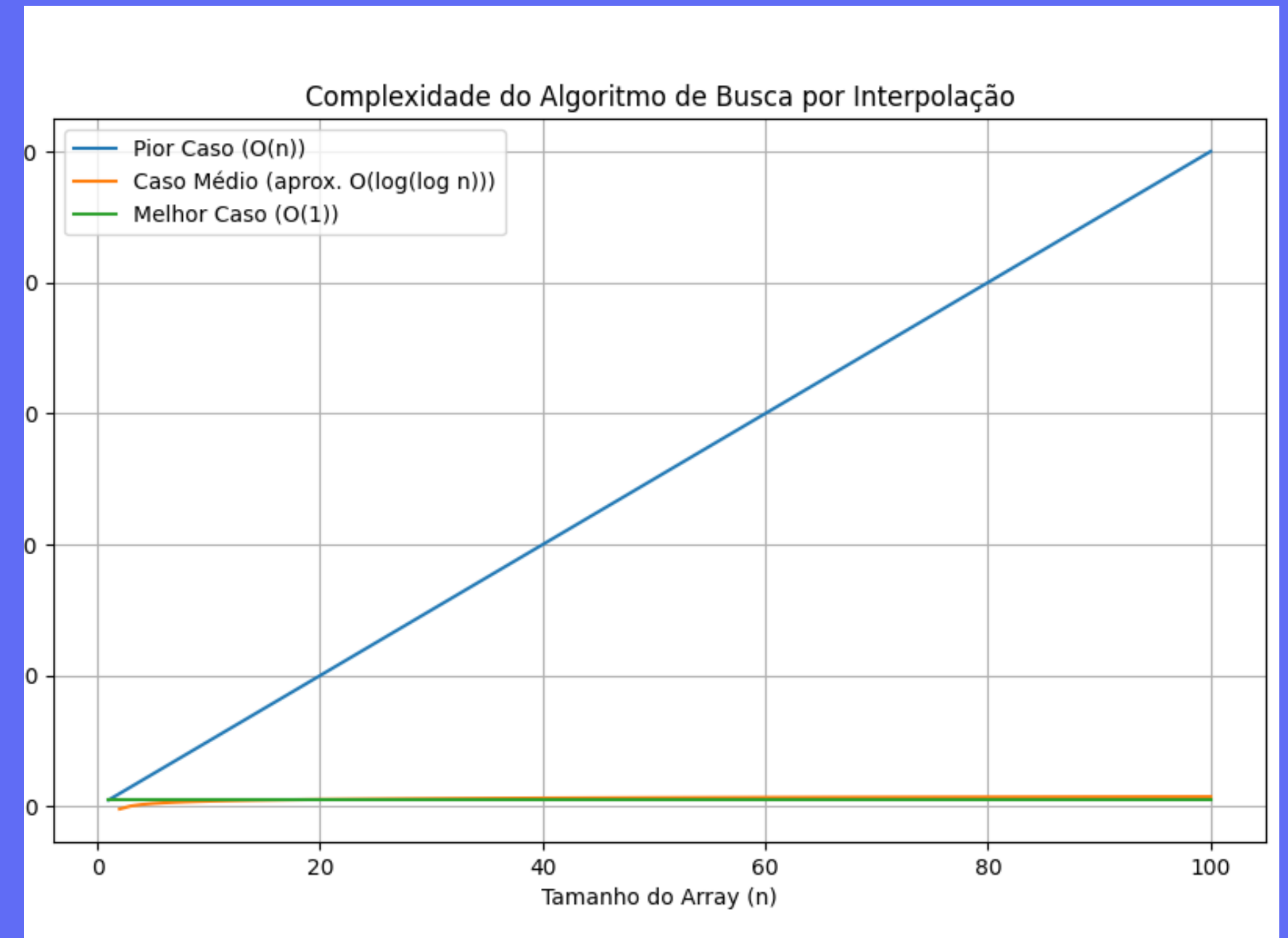
Complexidade de tempo:

- O melhor caso do algoritmo de Busca por Interpolação ocorre quando o elemento procurado é o primeiro elemento considerado. complexidade $O(1)$

$$c1 * 1 + c2 * 1 + c3 * 1 + c4 * 1 + c5 * 1 + c6 * 1 + c7 * 1 + c8 * 1 + c9 * 1 = O(1)$$

- O caso médio ocorre dependendo da uniformidade dos dados se assumirmos que os dados estão uniformemente distribuídos, a complexidade média é aproximadamente $O(\log(\log n))$ ou melhor.
- O pior caso do algoritmo de Busca por Interpolação ocorre quando o arranjo de busca não possui uma distribuição uniforme dos elementos. Nesse cenário, o algoritmo tem um custo linear de comparações. $O(n)$

$$c1 * 1 + c2 * 1 + c3 * n + c4 * n + c5 * n + c6 * n + c7 * n + c8 * n + c9 * 1 = O(n)$$



Complexidade de espaço:

A complexidade de espaço do algoritmo de busca por interpolação é $O(1)$. Isso ocorre porque o algoritmo geralmente não requer armazenamento adicional significativo além das variáveis locais e dos elementos de dados a serem pesquisados.



Vantagens

- **Eficiência em dados uniformemente distribuídos.**
- **Desempenho em grandes conjuntos de dados.**

Desvantagens

- **Desempenho irregular com distribuição não uniforme.**
- **Complexidade de implementação.**

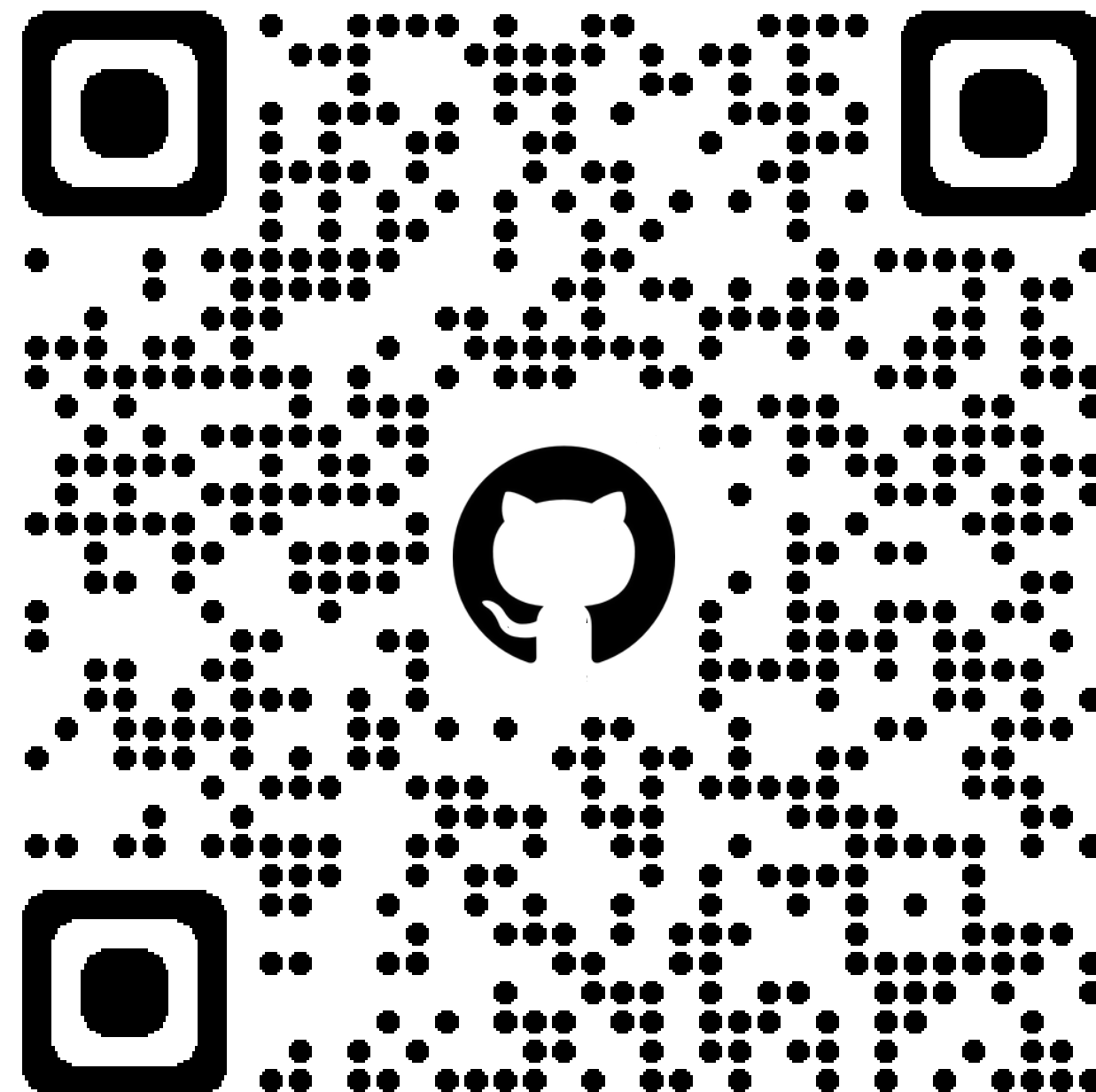


DEV

Repertório



Colaboradores





**como diria minha
ex, terminamos**

