



Universidade de Brasília
Faculdade UnB Gama
Disciplina: Estrutura de Dados e Algoritmos - EDA

Buscas

Algoritmos e Complexidades

Prof. Nilton Correia da Silva

16 de agosto de 2022

Introdução

- Conceitos Iniciais

- Métodos que Abordaremos

Busca Sequencial

- Vetor Desordenado

- Vetor Ordenado

Busca Binária

- Algoritmo

- Exemplos

Complexidade

Aplicação

Busca

Recuperar um dado que está armazenado em um conjunto de dados (por exemplo: banco de dados, arquivo ou vetor). A apresentação ordenada ou desordenada deste conjunto de dados impacta no tipo de busca aplicável.

Tipos de busca

- ▶ Dados desordenados: busca sequencial.
- ▶ Dados ordenados: busca binária.

Busca em Memória Principal

- ▶ Dados estão armazenados em um vetor.
- ▶ Vetor pode estar ordenado ou desordenado.

Algoritmos

- ▶ Busca linear em vetor desordenado.
- ▶ Busca linear em vetor ordenado.
- ▶ Busca binária em vetor ordenado.

```
1 //Busca chave no vetor desordenado V[0..n-1]
2 int buscaSequencialDesordenada(int *V, int n, int chave){
3     int p, posic = -1;
4     for(p=0;p<n;p++){
5         if(chave == V[p]){
6             posic = p;
7         }
8     }
9     return posic;
10 }
```

No caso de *chave* estar repetida em *V*?

- ▶ Qual delas a solução acima retorna?
- ▶ O que muda se adicionarmos *break;* na linha 7?

```
1 //Busca chave no vetor ordenado V[0..n-1]
2 int buscaSequencialOrdenada(int *V, int n, int chave) {
3     int p=0, posic = -1;
4     while (p < n && chave <= V[p]){
5         if(V[p]==chave){
6             posic = p;
7             break;
8         }
9         p++;
10    }
11    return posic;
12 }
```

No caso de *chave* estar repetida em *V*?

- ▶ Como ter uma solução que retorna a posição da última ocorrência de *chave*?

```
1  //Busca chave no vetor ordenado V[0..n-1]
2  int buscaBinaria(int *V, int n, int chave){
3      int p, inicio, final, meio;
4      inicio = 0;
5      final = n - 1;
6      while(inicio <= final){
7          meio = (inicio + final)/2;
8          if (chave == V[meio])
9              return meio;
10         if (chave < V[meio])
11             final = meio - 1; //busca nos valores menores
12         else
13             inicio = meio + 1; //busca nos valores maiores
14     }
15     return -1; //chave não encontrada
16 }
```

Vetor ordenado V[8]

buscaBinaria(V, 8, 84)

meio = 3

17	27	32	49	53	63	84	91
----	----	----	----	----	----	----	----

Inicio = 0
final = 7

meio = 5

17	27	32	49	53	63	84	91
---------------	---------------	---------------	---------------	---------------	---------------	----	----

Inicio = 4
final = 7

meio = 6

17	27	32	49	53	63	84	91
---------------	---------------	---------------	---------------	---------------	---------------	----	----

Inicio = 6
final = 7

6

Vetor ordenado V[8]

buscaBinaria(V, 8, 17)

meio = 3

17	27	32	49	53	63	84	91
----	----	----	----	----	----	----	----

Inicio = 0
final = 7

meio = 1

17	27	32	49	53	63	84	91
----	----	----	----	----	----	----	----

Inicio = 0
final = 2

meio = 0

17	27	32	49	53	63	84	91
----	----	----	----	----	----	----	----

Inicio = 0
final = 0

0

Vetor ordenado V[8]

buscaBinaria(V, 8, 64)

meio = 3

17	27	32	49	53	63	84	91
----	----	----	----	----	----	----	----

Início = 0
final = 7

meio = 5

17	27	32	49	53	63	84	91
---------------	---------------	---------------	----	---------------	----	----	----

Início = 4
final = 7

meio = 6

17	27	32	49	53	63	84	91
---------------	---------------	---------------	----	---------------	---------------	----	----

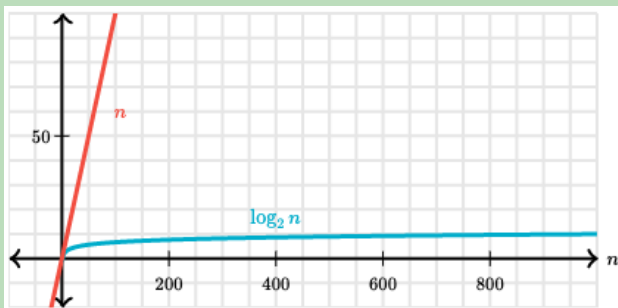
Início = 6
final = 7

-1

Início = 6
final = 5

Busca Sequencial: $O(n)$. Busca Binária: $O(\log_2 n)$

n	$\log_2 n$
1	0
1024	10
2.097.152	21



Enunciado

O arquivo de dados do censo 2010 do IBGE está ordenado pelo código de município (campo IBGE). Faça um programa com as seguintes opções:

1. Faça a carga de um vetor contendo **IBGE** e **Linha**. onde **Linha** refere-se ao número da linha do arquivo.
2. Leia um código de município e mostre na tela as seguintes informações deste município: **Município, UF, Região, População em 2010 e Porte**. Mostre ainda o tempo de execução desta busca com **Busca Binária** e **Busca Sequencial**.
3. Sair. Não deixe lixo na memória.

Dados

[http://blog.mds.gov.br/redesuas/
lista-de-municipios-brasileiros/](http://blog.mds.gov.br/redesuas/lista-de-municipios-brasileiros/)