



BY

“Maus programadores se preocupam com o código. Bons programadores se preocupam com as estruturas de dados e seus relacionamentos”  
(Linus Torvalds).

# Dicionários

Paulo Ricardo Lisboa de Almeida



Baseado nas aulas do Professor Eduardo Almeida - [www.inf.ufpr.br/eduardo](http://www.inf.ufpr.br/eduardo)



# Dicionário

Muitas vezes desejamos armazenar elementos que possam ser encontrados por uma chave de busca.

Exemplo: em um sistema do governo, desejamos encontrar os dados de uma pessoa a partir de seu CPF.

Mais exemplos?

# Dicionário

Muitas vezes desejamos armazenar elementos que possam ser encontrados por uma chave de busca.

- Em um sistema do governo, desejamos encontrar os dados de uma pessoa a partir de seu CPF.
- Em um sistema do Detran, desejamos encontrar os dados do veículo pela placa.
- No seu banco (BB, Itaú, ...), os sistemas buscam suas informações a partir do número da sua conta.
- Na UFPR, suas informações são atreladas ao seu GRR.
- ...

# Lista Versus Dicionário

Lista	Dicionário
Sequência ordenada de elementos.	Relaciona Chave com Valor.
Busca item por índice.	Busca item a partir de uma chave.
Índices tem ordem.	Sem ordem específica.
Índices são do tipo inteiro.	Chave pode ter qualquer tipo.

0	1	2	...
Gol Cinza 2013 Placa AAA-1G11	Accord Branco 2022 Placa BBB-2B22	Uno Azul 2015 Placa CCC-3D33	...



# Lista Versus Dicionário

Lista	Dicionário
Sequência ordenada de elementos.	Relaciona Chave com Valor.
Busca item por índice.	Busca item a partir de uma chave.
Índices tem ordem.	Sem ordem específica.
Índices são do tipo inteiro.	Chave pode ter qualquer tipo.
Busque o item no índice 1.	Busque o item de placa BBB-2B222.

Exemplo de Operação ->

0                      1                      2                      ...

Gol Cinza 2013 Placa AAA-1G11	Accord Branco 2022 Placa BBB-2B22	Uno Azul 2015 Placa CCC-3D33	...
----------------------------------	--------------------------------------	---------------------------------	-----

Gol Cinza 2013  
Placa AAA-1G11

Uno Azul 2015  
Placa CCC-3D33

Accord Branco  
Placa BBB-2B22

...

# Bibliotecas prontas

Estruturas de dicionário são tão úteis que a maioria das linguagens as possuem em suas bibliotecas padrão.

Exemplos:

`unordered_map` em C++;

`GHashTable` em C (lib GLib);

`HashMap` em Java;

`Dictionary` na plataforma .Net (ex.: C#);

`Dictionary` em Python.

# Bibliotecas prontas

Estruturas de dicionário são tão úteis que a maioria das linguagens as possuem em suas bibliotecas padrão.

Exemplos:

- `unordered_map` em C++;
- `GHashTable` em C (lib GLib);
- `HashMap` em Java;
- `Dictionary` na plataforma .Net (ex.: C#);
- `Dictionary` em Python.

Exemplo em Python.

```
dict = {}  
dict["AAA-1G11"] = ("Gol", "Cinza", 2013)  
dict["BBB-2B22"] = ("Accord", "Branco", 2022)  
dict["CCC-3D33"] = ("Uno", "Azul", 2015)
```

```
print(dict["BBB-2B22"])
```

**Console**  
( 'Accord', 'Branco', 2022)

# Operações

Geralmente em um dicionário são feitas as seguintes operações:

```
item buscar(chave);  
bool inserir(item);  
item excluir(item);
```



# Representando um item

Como representar um item (exemplo: informações do carro) em C?

# Representando um item

Como representar um item (exemplo: informações do carro) em C?

Usamos uma `struct`.

De maneira geral (independente de linguagem), vamos definir um tipo heterogêneo de dados.

- Uma `struct` em C.
- `Record` em Pascal.
- `Class` em C++.
- ...

# TAD Dicionário

Precisamos criar um tipo abstrato de dados (TAD) para armazenar os itens.

Diversas formas:

- Arquivo de log
- Dicionário ordenado
- Tabela Hash
- SkipList
- ...

# TAD Dicionário

Armazena pares (chave, valor).

$(k, v)$ .

Pares chave e valor podem ser de qualquer tipo.

Exemplo: a chave é uma string, e o valor é uma struct.

Pares chave e valor podem ser inclusive iguais.

AAA-1G111

BBB-2B222

CCC-3D333

...

Gol Cinza 2013 Placa AAA-1G11	Accord Branco 2022 Placa BBB-2B22	Uno Azul 2015 Placa CCC-3D33	...
----------------------------------	--------------------------------------	---------------------------------	-----

# TAD Dicionário

Exemplo em C.

```
struct carro{
    char modelo[MAX_CHAR];
    unsigned short int ano;
    char placa[TAM_PLACA];
};

struct dicionario_carro{
    struct carro* carros[TAM_DICT];
    unsigned long numItens;
};

void inicializar(struct dicionario_carro* dict){
    dict->numItens = 0;
}

struct carro* buscar(char* chave, struct dicionario_carro* dict);
int inserir(struct carro* carro, struct dicionario_carro* dict);
struct carro* remover(char* chave, struct dicionario_carro* dict);
```

# Ordenado ou não ordenado

Ordenado:

- Ordem relativa determinada pelo comparador de chaves;
- Ordem total definida pelas chaves.

Não Ordenado:

- Não assumimos nenhuma ordem relativa;
- Somente validação de igualdade pela chave.
  - Podemos apenas perguntar se a chave de determinado item é igual a determinada chave.

# TAD Dicionário: Arquivo de log

TAD Dicionário não ordenado para armazenar mensagens.

Útil quando temos muitas inclusões e poucas buscas.

Pode ser implementado num vetor ou lista encadeada.

Aplicações: Bancos de dados, sistemas operacionais, sistemas de auditoria, ...

# Exemplo

```
cat /var/log/syslog
```

```
Feb 24 11:34:37 paulo-desk rtkit-daemon[1114]: Supervising 9 threads of 4 processes of 1 users.  
Feb 24 11:34:39 paulo-desk rtkit-daemon[1114]: message repeated 3 times: [ Supervising 9 threads of 4 processes of 1  
users.]  
Feb 24 11:34:41 paulo-desk systemd[1]: NetworkManager-dispatcher.service: Succeeded.  
Feb 24 11:35:18 paulo-desk systemd[1]: Started Run anacron jobs.  
Feb 24 11:35:18 paulo-desk anacron[27708]: Anacron 2.3 started on 2023-02-24  
Feb 24 11:35:18 paulo-desk anacron[27708]: Normal exit (0 jobs run)  
Feb 24 11:35:18 paulo-desk systemd[1]: anacron.service: Succeeded.  
Feb 24 11:35:36 paulo-desk rtkit-daemon[1114]: Supervising 9 threads of 4 processes of 1 users.  
Feb 24 11:36:21 paulo-desk rtkit-daemon[1114]: message repeated 7 times: [ Supervising 9 threads of 4 processes of 1  
users.]  
Feb 24 11:38:37 paulo-desk rtkit-daemon[1114]: Supervising 8 threads of 3 processes of 1 users.
```



# TAD Dicionário: Dicionário Ordenado

TAD Dicionário ordenado para armazenar para busca binária.

Útil quando temos muitas buscas e poucas inclusões/exclusões.

Pode ser implementado em um vetor.

# Dica de programação

Qual o tamanho máximo (teórico) de um vetor? E de um malloc?

Veja em [prl Almeida.com.br/knowledge-base/use-size\\_t](http://prl Almeida.com.br/knowledge-base/use-size_t)

Veja sobre o `size_t` também na Especificação ISO/IEC 9899:2018.

7.19 Common definitions `<stddef.h>`

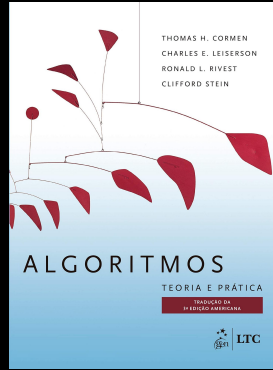
7.20.3 Limits of other integer types

# Exercícios

1. Crie uma struct para representar uma pessoa.
2. Crie um dicionário (não ordenado) de pessoas. A chave é o cpf. Use o exemplo dos carros dado em aula.
3. Crie as funções necessárias em C (buscar, inserir, ...). Faça um main para testes.
4. Modifique o programa para que o dicionário seja ordenado por chave.
  - a. A função de inserção deve inserir no local correto.
  - b. A função de busca deve realizar uma busca binária.

# Referências

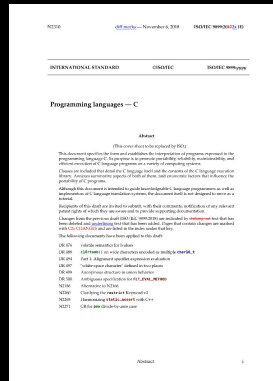
T. Cormen, C. Leiserson, R. Rivest, C. Stein. Algoritmos: Teoria e Prática. 3a ed. 2012.



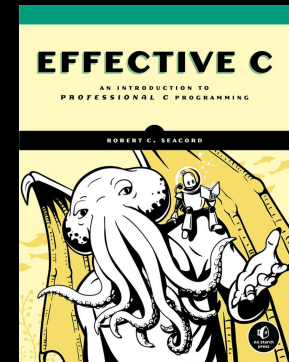
R. Sedgwick, K. Wayne. Algorithms Part I. 4a ed. 2011



ISO/IEC 9899:2018



Seacord, R. C. Effective C: An introduction to Professional C Programming. 2020.



# Licença

Esta obra está licenciada com uma Licença [Creative Commons Atribuição 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).