

Estruturas de Dados

Hashing

Aula 14

Prof. Felipe A. Louza

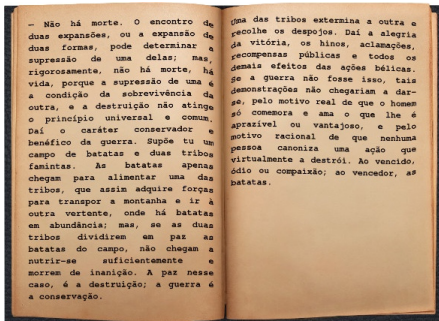


- 1 Introdução
- 2 Função de hashing
- 3 Hashing com encadeamento
- 4 Hashing com endereçamento aberto
- 5 Referências

- 1 Introdução
- 2 Função de hashing
- 3 Hashing com encadeamento
- 4 Hashing com endereçamento aberto
- 5 Referências

Introdução

Suponha que queremos saber o **número de ocorrências** de **cada palavra** em uma biblioteca



- no idioma, há cerca de **milhares de palavras** (≈ 200.000)
- no total, podemos ter **milhões de ocorrências**!

Exemplo

"dia":	6 ocorrências
"escola":	13 ocorrências
"gratuito":	1 ocorrência
"ilha":	8 ocorrências
"jeito":	5 ocorrências
"lata":	2 ocorrências

Exemplo

"dia":	6 ocorrências
"escola":	13 ocorrências
"gratuito":	1 ocorrência
"ilha":	8 ocorrências
"jeito":	5 ocorrências
"lata":	2 ocorrências

Queremos acessar uma palavra como se fosse um vetor:

```
ocorrencias["ilha"] = 8
```

Exemplo

"dia":	6 ocorrências
"escola":	13 ocorrências
"gratuito":	1 ocorrência
"ilha":	8 ocorrências
"jeito":	5 ocorrências
"lata":	2 ocorrências

Queremos acessar uma palavra como se fosse um vetor:

```
ocorrencias["ilha"] = 8
```

Primeiras opções:

Exemplo

"dia":	6 ocorrências
"escola":	13 ocorrências
"gratuito":	1 ocorrência
"ilha":	8 ocorrências
"jeito":	5 ocorrências
"lata":	2 ocorrências

Queremos acessar uma palavra como se fosse um vetor:

`ocorrencias["ilha"] = 8`

Primeiras opções:

- Vetor - acesso/escrita em $O(n)$

Exemplo

"dia":	6 ocorrências
"escola":	13 ocorrências
"gratuito":	1 ocorrência
"ilha":	8 ocorrências
"jeito":	5 ocorrências
"lata":	2 ocorrências

Queremos acessar uma palavra como se fosse um vetor:

`ocorrencias["ilha"] = 8`

Primeiras opções:

- Vetor - acesso/escrita em $O(n)$
 - inserir uma nova palavra leva $O(1)$

Exemplo

"dia":	6 ocorrências
"escola":	13 ocorrências
"gratuito":	1 ocorrência
"ilha":	8 ocorrências
"jeito":	5 ocorrências
"lata":	2 ocorrências

Queremos acessar uma palavra como se fosse um vetor:

`ocorrencias["ilha"] = 8`

Primeiras opções:

- Vetor - acesso/escrita em $O(n)$
 - inserir uma nova palavra leva $O(1)$
- Vetor ordenado - acesso/escrita em $O(\lg n)$

Exemplo

"dia":	6 ocorrências
"escola":	13 ocorrências
"gratuito":	1 ocorrência
"ilha":	8 ocorrências
"jeito":	5 ocorrências
"lata":	2 ocorrências

Queremos acessar uma palavra como se fosse um vetor:

`ocorrencias["ilha"] = 8`

Primeiras opções:

- Vetor - acesso/escrita em $O(n)$
 - inserir uma nova palavra leva $O(1)$
- Vetor ordenado - acesso/escrita em $O(\lg n)$
 - inserir uma nova palavra leva $O(n)$

Exemplo

"dia":	6 ocorrências
"escola":	13 ocorrências
"gratuito":	1 ocorrência
"ilha":	8 ocorrências
"jeito":	5 ocorrências
"lata":	2 ocorrências

Queremos acessar uma palavra como se fosse um vetor:

`ocorrencias["ilha"] = 8`

Primeiras opções:

- Vetor - acesso/escrita em $O(n)$
 - inserir uma nova palavra leva $O(1)$
- Vetor ordenado - acesso/escrita em $O(\lg n)$
 - inserir uma nova palavra leva $O(n)$
- ABB balanceada - acesso/escrita e inserção em $O(\lg n)$

Exemplo

"dia":	6 ocorrências
"escola":	13 ocorrências
"gratuito":	1 ocorrência
"ilha":	8 ocorrências
"jeito":	5 ocorrências
"lata":	2 ocorrências

Queremos acessar uma palavra como se fosse um vetor:

`ocorrencias["ilha"] = 8`

Primeiras opções:

- Vetor - acesso/escrita em $O(n)$
 - inserir uma nova palavra leva $O(1)$
- Vetor ordenado - acesso/escrita em $O(\lg n)$
 - inserir uma nova palavra leva $O(n)$
- ABB balanceada - acesso/escrita e inserção em $O(\lg n)$

Conseguimos fazer em $O(1)$?

Caso fácil

Se tivéssemos apenas uma palavra começando com cada letra seria fácil

a	0	
b	1	
c	2	
d	3	dia
e	4	escola
f	5	
g	6	gratuito
h	7	
i	8	ilha
⋮	⋮	⋮

Caso fácil

Se tivéssemos apenas uma palavra começando com cada letra seria fácil

- bastaria ter um vetor de 26 posições

a	0	
b	1	
c	2	
d	3	dia
e	4	escola
f	5	
g	6	gratuito
h	7	
i	8	ilha
⋮	⋮	⋮

Caso fácil

Se tivéssemos apenas uma palavra começando com cada letra seria fácil

- bastaria ter um vetor de 26 posições
- acesso/escrita e inserção em $O(1)$

a	0	
b	1	
c	2	
d	3	dia
e	4	escola
f	5	
g	6	gratuito
h	7	
i	8	ilha
⋮	⋮	⋮

Palavras que começam com a mesma letra

0	→	NULL
1	→	NULL
2	→	NULL
3	→	dia
4	→	escola
5	→	NULL
6	→	gratuito
7	→	NULL
8	→	ilha
⋮	⋮	⋮

Palavras que começam com a mesma letra

0	→	NULL
1	→	NULL
2	→	NULL
3	→	dia
4	→	escola
5	→	NULL
6	→	gratuito
7	→	NULL
8	→	ilha
⋮	⋮	⋮

Ideia:

Palavras que começam com a mesma letra



Ideia:

- uma **lista ligada** para cada letra

Palavras que começam com a mesma letra



Ideia:

- uma **lista ligada** para cada letra
- guardamos os ponteiros para as listas em um vetor

Palavras que começam com a mesma letra

0	→	NULL
1	→	NULL
2	→	NULL
3	→	dia
4	→	escola
5	→	NULL
6	→	gratuito
7	→	NULL
8	→	ilha
⋮	⋮	⋮

Palavras que começam com a mesma letra

0	→	NULL
1	→	NULL
2	→	NULL
3	→	dia
4	→	escola
5	→	NULL
6	→	gratuito
7	→	NULL
8	→	ilha
⋮	⋮	⋮

Inserindo “bala”:

Palavras que começam com a mesma letra

0	→	NULL
1	→	NULL
2	→	NULL
3	→	dia
4	→	escola
5	→	NULL
6	→	gratuito
7	→	NULL
8	→	ilha
⋮	⋮	⋮

Inserindo “bala”:

- descobrimos a **posição** pela primeira letra

Palavras que começam com a mesma letra

0	→	NULL
1	→	NULL
2	→	NULL
3	→	dia
4	→	escola
5	→	NULL
6	→	gratuito
7	→	NULL
8	→	ilha
⋮	⋮	⋮

Inserindo “bala”:

- descobrimos a **posição** pela primeira letra
- **atualizamos** o vetor para apontar para o nó de “bala”

Palavras que começam com a mesma letra

0	→	NULL
1	→	bala
2	→	NULL
3	→	dia
4	→	escola
5	→	NULL
6	→	gratuito
7	→	NULL
8	→	ilha
⋮	⋮	⋮

Inserindo “bala”:

- descobrimos a **posição** pela primeira letra
- **atualizamos** o vetor para apontar para o nó de “bala”

Palavras que começam com a mesma letra

0	→	NULL
1	→	bala
2	→	NULL
3	→	dia
4	→	escola
5	→	NULL
6	→	gratuito
7	→	NULL
8	→	ilha
⋮	⋮	⋮

Palavras que começam com a mesma letra

0	→	NULL
1	→	bala
2	→	NULL
3	→	dia
4	→	escola
5	→	NULL
6	→	gratuito
7	→	NULL
8	→	ilha
⋮	⋮	⋮

Inserindo “bela”:

Palavras que começam com a mesma letra

0	→	NULL
1	→	bala
2	→	NULL
3	→	dia
4	→	escola
5	→	NULL
6	→	gratuito
7	→	NULL
8	→	ilha
⋮	⋮	⋮

Inserindo “bela”:

- descobrimos a **posição** pela primeira letra

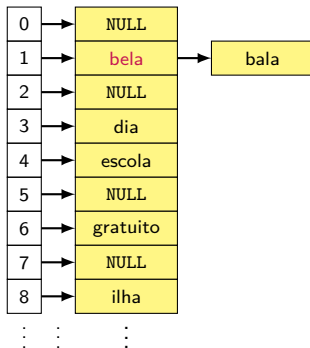
Palavras que começam com a mesma letra

0	→	NULL
1	→	bala
2	→	NULL
3	→	dia
4	→	escola
5	→	NULL
6	→	gratuito
7	→	NULL
8	→	ilha
⋮	⋮	⋮

Inserindo “bela”:

- descobrimos a **posição** pela primeira letra
- temos uma **colisão** com “bala”

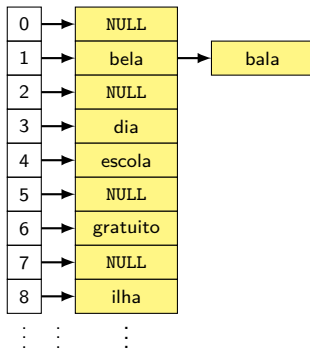
Palavras que começam com a mesma letra



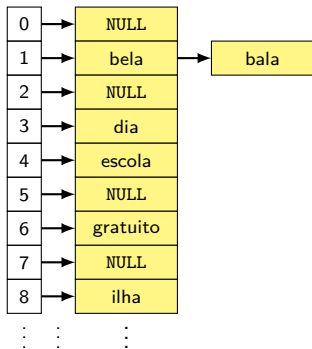
Inserindo “bela”:

- descobrimos a **posição** pela primeira letra
- temos uma **colisão** com “bala”
- inserimos no começo da lista da letra **b**

Palavras que começam com a mesma letra

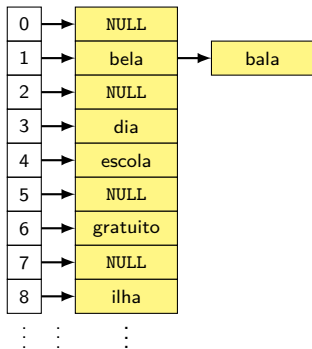


Palavras que começam com a mesma letra



Após a **inserção** de **várias palavras** começando com **b**:

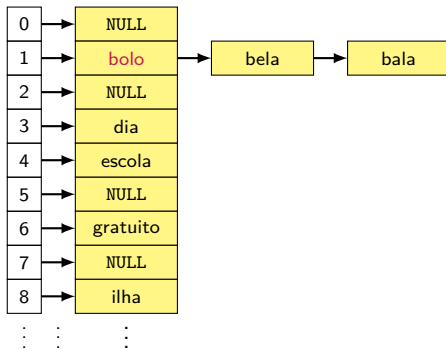
Palavras que começam com a mesma letra



Após a **inserção** de **várias palavras** começando com **b**:

- inserimos “bolo”,

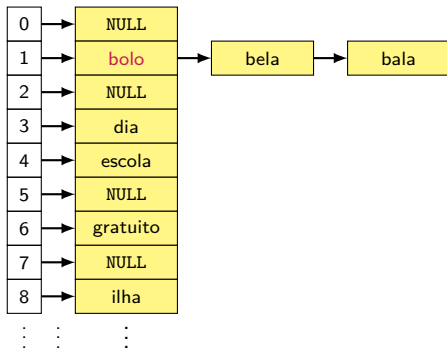
Palavras que começam com a mesma letra



Após a **inserção** de **várias palavras** começando com **b**:

- inserimos “bolo”,

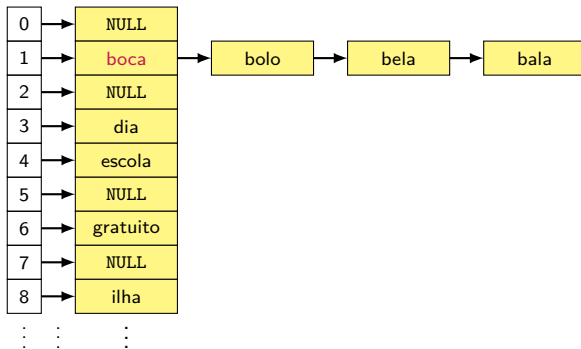
Palavras que começam com a mesma letra



Após a **inserção** de **várias palavras** começando com **b**:

- inserimos “bolo”, “boca”,

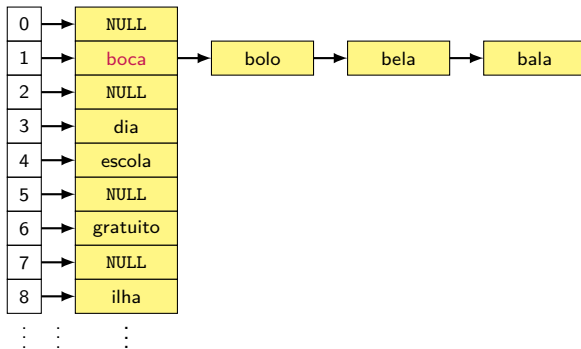
Palavras que começam com a mesma letra



Após a **inserção** de **várias palavras** começando com **b**:

- inserimos “bolo”, “boca”,

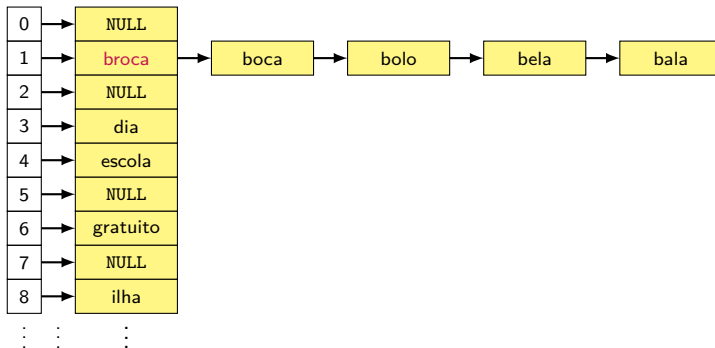
Palavras que começam com a mesma letra



Após a **inserção** de **várias palavras** começando com **b**:

- inserimos “bolo”, “boca”, “broca”

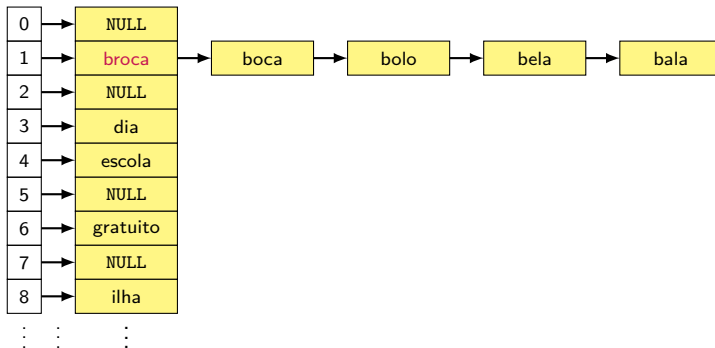
Palavras que começam com a mesma letra



Após a **inserção** de **várias palavras** começando com **b**:

- inserimos “bolo”, “boca”, “broca”

Palavras que começam com a mesma letra



Após a **inserção** de **várias palavras** começando com **b**:

- inserimos “bolo”, “boca”, “broca”
- a tabela ficou degenerada em lista

Espalhamento com Encadeamento Separado

broca

boca

bolo

bela

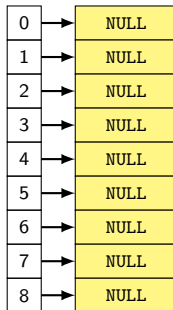
bala

dia

escola

gratuito

ilha



Corrigindo:

Espalhamento com Encadeamento Separado

broca

boca

bolo

bela

bala

dia

escola

gratuito

ilha

0	→	NULL
1	→	NULL
2	→	NULL
3	→	NULL
4	→	NULL
5	→	NULL
6	→	NULL
7	→	NULL
8	→	NULL

Corrigindo:

- vamos tentar **espalhar** melhor

Espalhamento com Encadeamento Separado

broca

boca

bolo

bela

bala

dia

escola

gratuito

ilha

0	→	NULL
1	→	NULL
2	→	NULL
3	→	NULL
4	→	NULL
5	→	NULL
6	→	NULL
7	→	NULL
8	→	NULL

Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um hash da chave (palavra)

Espalhamento com Encadeamento Separado

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca

bolo

bela

bala

dia

escola

gratuito

ilha

0	→	NULL
1	→	NULL
2	→	NULL
3	→	NULL
4	→	NULL
5	→	NULL
6	→	NULL
7	→	NULL
8	→	NULL

Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um **hash** da chave (palavra)
- vamos **associar a chave** a um **número inteiro** (entre 0 e 8)

Espalhamento com Encadeamento Separado

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca

bolo

bela

bala

dia

escola

gratuito

ilha

0	→	NULL
1	→	NULL
2	→	NULL
3	→	broca
4	→	NULL
5	→	NULL
6	→	NULL
7	→	NULL
8	→	NULL

Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um **hash** da chave (palavra)
- vamos **associar a chave** a um **número inteiro** (entre 0 e 8)

Espalhamento com Encadeamento Separado

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo

bela

bala

dia

escola

gratuito

ilha

0	→	NULL
1	→	NULL
2	→	NULL
3	→	broca
4	→	NULL
5	→	NULL
6	→	NULL
7	→	NULL
8	→	NULL

Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um **hash** da chave (palavra)
- vamos **associar a chave** a um **número inteiro** (entre 0 e 8)

Espalhamento com Encadeamento Separado

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo

bela

bala

dia

escola

gratuito

ilha

0	→	boca
1	→	NULL
2	→	NULL
3	→	broca
4	→	NULL
5	→	NULL
6	→	NULL
7	→	NULL
8	→	NULL

Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um **hash** da chave (palavra)
- vamos **associar a chave** a um **número inteiro** (entre 0 e 8)

Espalhamento com Encadeamento Separado

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela

bala

dia

escola

gratuito

ilha

0	→	boca
1	→	NULL
2	→	NULL
3	→	broca
4	→	NULL
5	→	NULL
6	→	NULL
7	→	NULL
8	→	NULL

Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um **hash** da chave (palavra)
- vamos **associar a chave** a um **número inteiro** (entre 0 e 8)

Espalhamento com Encadeamento Separado

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela

bala

dia

escola

gratuito

ilha

0	→	boca
1	→	NULL
2	→	NULL
3	→	broca
4	→	NULL
5	→	bolo
6	→	NULL
7	→	NULL
8	→	NULL

Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um **hash** da chave (palavra)
- vamos **associar a chave** a um **número inteiro** (entre 0 e 8)

Espalhamento com Encadeamento Separado

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala

dia

escola

gratuito

ilha

0	→	boca
1	→	NULL
2	→	NULL
3	→	broca
4	→	NULL
5	→	bolo
6	→	NULL
7	→	NULL
8	→	NULL

Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um **hash** da chave (palavra)
- vamos **associar a chave** a um **número inteiro** (entre 0 e 8)

Espalhamento com Encadeamento Separado

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala

dia

escola

gratuito

ilha

0	→	boca
1	→	NULL
2	→	bela
3	→	broca
4	→	NULL
5	→	bolo
6	→	NULL
7	→	NULL
8	→	NULL

Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um **hash** da chave (palavra)
- vamos **associar a chave** a um **número inteiro** (entre 0 e 8)

Espalhamento com Encadeamento Separado

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia

escola

gratuito

ilha

0	→	boca
1	→	NULL
2	→	bela
3	→	broca
4	→	NULL
5	→	bolo
6	→	NULL
7	→	NULL
8	→	NULL

Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um **hash** da chave (palavra)
- vamos **associar a chave** a um **número inteiro** (entre 0 e 8)

Espalhamento com Encadeamento Separado

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

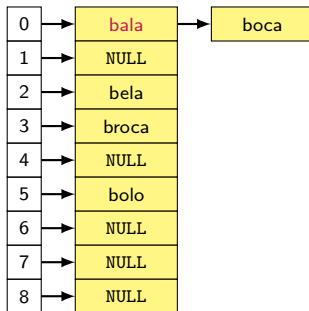
bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia

escola

gratuito

ilha



Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um **hash** da chave (palavra)
- vamos **associar a chave** a um **número inteiro** (entre 0 e 8)

Espalhamento com Encadeamento Separado

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

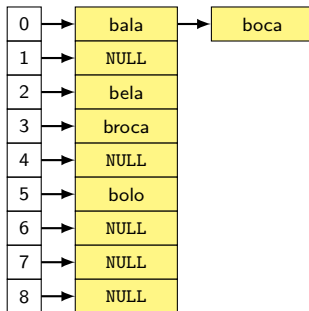
bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola

gratuito

ilha



Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um **hash** da chave (palavra)
- vamos **associar a chave** a um **número inteiro** (entre 0 e 8)

Espalhamento com Encadeamento Separado

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

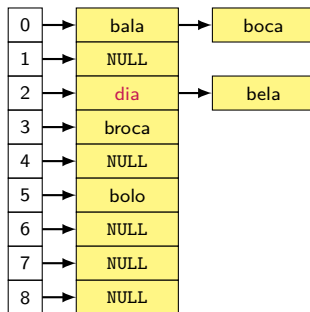
bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola

gratuito

ilha



Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um **hash** da chave (palavra)
- vamos **associar a chave** a um **número inteiro** (entre 0 e 8)

Espalhamento com Encadeamento Separado

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

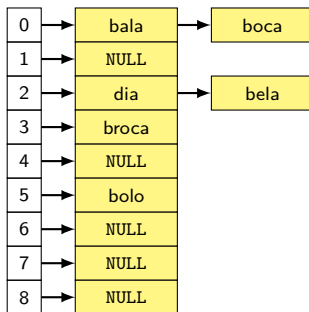
bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola $\rightsquigarrow h(\text{"escola"}) = 7$

gratuito

ilha



Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um **hash** da chave (palavra)
- vamos **associar a chave** a um **número inteiro** (entre 0 e 8)

Espalhamento com Encadeamento Separado

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

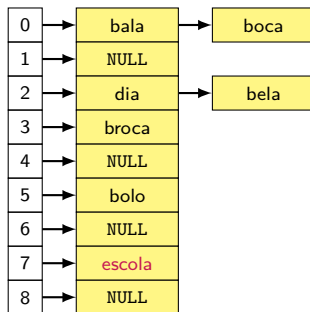
bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola $\rightsquigarrow h(\text{"escola"}) = 7$

gratuito

ilha



Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um **hash** da chave (palavra)
- vamos **associar a chave** a um **número inteiro** (entre 0 e 8)

Espalhamento com Encadeamento Separado

broca \rightsquigarrow $h(\text{"broca"}) = 3$

boca \rightsquigarrow $h(\text{"boca"}) = 0$

bolo \rightsquigarrow $h(\text{"bolo"}) = 5$

bela \rightsquigarrow $h(\text{"bela"}) = 2$

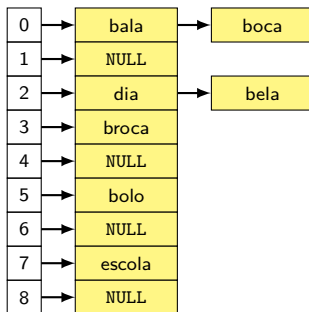
bala \rightsquigarrow $h(\text{"bala"}) = 0$

dia \rightsquigarrow $h(\text{"dia"}) = 2$

escola \rightsquigarrow $h(\text{"escola"}) = 7$

gratuito \rightsquigarrow $h(\text{"gratuito"}) = 0$

ilha



Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um **hash** da chave (palavra)
- vamos **associar a chave** a um **número inteiro** (entre 0 e 8)

Espalhamento com Encadeamento Separado

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

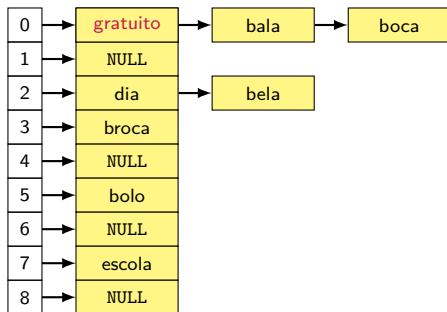
bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola $\rightsquigarrow h(\text{"escola"}) = 7$

gratuito $\rightsquigarrow h(\text{"gratuito"}) = 0$

ilha



Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um **hash** da chave (palavra)
- vamos **associar a chave** a um **número inteiro** (entre 0 e 8)

Espalhamento com Encadeamento Separado

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

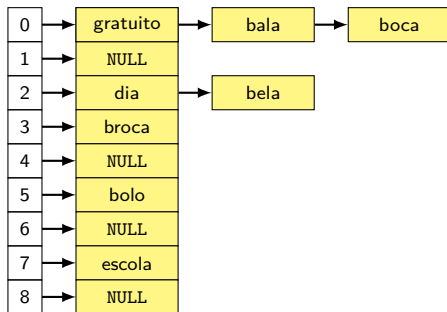
bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola $\rightsquigarrow h(\text{"escola"}) = 7$

gratuito $\rightsquigarrow h(\text{"gratuito"}) = 0$

ilha $\rightsquigarrow h(\text{"ilha"}) = 6$



Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um **hash** da chave (palavra)
- vamos **associar a chave** a um **número inteiro** (entre 0 e 8)

Espalhamento com Encadeamento Separado

broca \rightsquigarrow $h(\text{"broca"}) = 3$

boca \rightsquigarrow $h(\text{"boca"}) = 0$

bolo \rightsquigarrow $h(\text{"bolo"}) = 5$

bela \rightsquigarrow $h(\text{"bela"}) = 2$

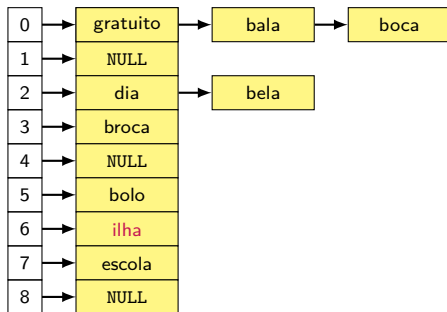
bala \rightsquigarrow $h(\text{"bala"}) = 0$

dia \rightsquigarrow $h(\text{"dia"}) = 2$

escola \rightsquigarrow $h(\text{"escola"}) = 7$

gratuito \rightsquigarrow $h(\text{"gratuito"}) = 0$

ilha \rightsquigarrow $h(\text{"ilha"}) = 6$



Corrigindo:

- vamos tentar **espalhar** melhor
- usamos um **hash** da chave (palavra)
- vamos **associar a chave** a um **número inteiro** (entre 0 e 8)

- 1 Introdução
- 2 Função de hashing**
- 3 Hashing com encadeamento
- 4 Hashing com endereçamento aberto
- 5 Referências

Tabela de Espalhamento

Uma função de hashing h associa

- um elemento de um conjunto de chaves \mathbb{C} (números, strings, etc.)
- a um **número natural** de tamanho conhecido

$$h : \mathbb{C} \rightarrow \mathbb{N}$$

Tabela de Espalhamento

Uma função de hashing h associa

- um elemento de um conjunto de chaves \mathbb{C} (números, strings, etc.)
- a um **número natural** de tamanho conhecido

$$h : \mathbb{C} \rightarrow \mathbb{N}$$

Uma tabela de espalhamento (*Hash table*) é uma **estrutura de dados dinâmica** em que:

- os **dados são acessado** por meio de um vetor de **tamanho** M
- a **posição do vetor** é calculada por uma função de hashing

Características das tabelas de espalhamento

Restrições

- **estimativa** do **tamanho do conjunto** de dados deve ser conhecida
- **bits da chave** devem estar disponíveis

Características das tabelas de espalhamento

Restrições

- **estimativa** do **tamanho do conjunto** de dados deve ser conhecida
- **bits da chave** devem estar disponíveis

Tempo das operações **depende** principalmente da função de hashing escolhida:

Características das tabelas de espalhamento

Restrições

- **estimativa** do **tamanho do conjunto** de dados deve ser conhecida
- **bits da chave** devem estar disponíveis

Tempo das operações **depende** principalmente da função de hashing escolhida:

- ① chaves bem espalhadas: tempo “quase” $O(1)$
 - se temos n itens
 - uma tabela de tamanho M
 - tempo de acesso é o **tempo de calcular** $h(\text{chave})$ + $O(n/M)$

Características das tabelas de espalhamento

Restrições

- **estimativa** do **tamanho do conjunto** de dados deve ser conhecida
- **bits da chave** devem estar disponíveis

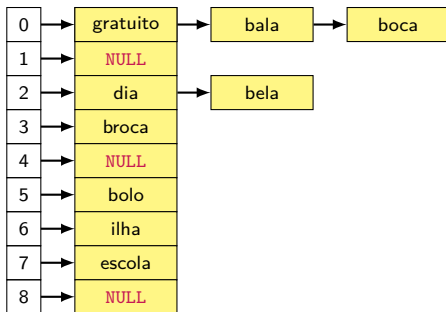
Tempo das operações **depende** principalmente da função de hashing escolhida:

- ① chaves bem espalhadas: tempo “quase” $O(1)$
 - se temos n itens
 - uma tabela de tamanho M
 - tempo de acesso é o **tempo de calcular** $h(\text{chave}) + O(n/M)$
- ② chaves muito agrupadas: pior caso de tempo $O(n)$
 - vira uma lista ligada com todos os elementos

Obtendo funções de hashing

Uma **boa função** de hashing deve **espalhar bem**:

- A probabilidade de uma chave ter um hash específico é $\approx 1/M$
- Ou seja, esperamos que cada lista tenha n/M elementos



Função de hashing modular

Método da divisão:

- obtemos o resto da divisão pelo tamanho M do hashing

$$h(x) = x \bmod M$$

Função de hashing modular

Método da divisão:

- obtemos o resto da divisão pelo tamanho M do hashing

$$h(x) = x \bmod M$$

Exemplos com $M = 97$:

$$h(212) = 212 \bmod 97 = 18$$

$$h(618) = 618 \bmod 97 = 36$$

$$h(302) = 302 \bmod 97 = 11$$

Função de hashing modular

Método da divisão:

- obtemos o resto da divisão pelo **tamanho M** do hashing

$$h(x) = x \bmod M$$

Exemplos com $M = 97$:

$$h(212) = 212 \bmod 97 = 18$$

$$h(618) = 618 \bmod 97 = 36$$

$$h(302) = 302 \bmod 97 = 11$$

Escolhendo M :

- Escolher M como 2^k **não** é uma boa ideia
 - considera apenas os k bits menos significativos, ex. $M = 16$

Função de hashing modular

Método da divisão:

- obtemos o resto da divisão pelo **tamanho M** do hashing

$$h(x) = x \bmod M$$

Exemplos com $M = 97$:

$$h(212) = 212 \bmod 97 = 18$$

$$h(618) = 618 \bmod 97 = 36$$

$$h(302) = 302 \bmod 97 = 11$$

Escolhendo M :

- Escolher M como 2^k **não** é uma boa ideia
 - considera apenas os k bits menos significativos, ex. $M = 16$
- Em geral, escolhemos M como um número primo **longe de** uma potência de 2

Função de hashing modular

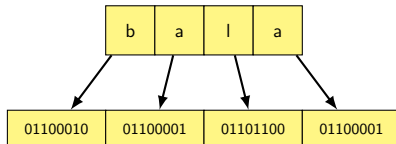
key	hash ($M = 100$)	hash ($M = 97$)
212	12	18
618	18	36
302	2	11
940	40	67
702	2	23
704	4	25
612	12	30
606	6	24
772	72	93
510	10	25
423	23	35
650	50	68
317	17	26
907	7	34
507	7	22
304	4	13
714	14	35
857	57	81
801	1	25
900	0	27
413	13	25
701	1	22
418	18	30
601	1	19

Modular hashing

Método da divisão para Strings

Como podemos obter um número x que representa a chave “bala”?

- Vamos considerar que “bala” é um “número” em base $b = 256$

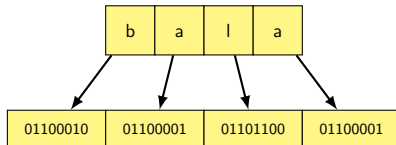


$$x = 'b' \cdot 256^3 + 'a' \cdot 256^2 + 'l' \cdot 256^1 + 'a' \cdot 256^0$$

Método da divisão para Strings

Como podemos obter um número x que representa a chave “bala”?

- Vamos considerar que “bala” é um “número” em base $b = 256$

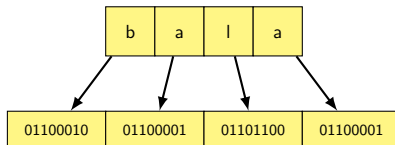


$$x = 'b' \cdot 256^3 + 'a' \cdot 256^2 + 'l' \cdot 256^1 + 'a' \cdot 256^0 = 1650551905$$

Método da divisão para Strings

Como podemos obter um número x que representa a chave “bala”?

- Vamos considerar que “bala” é um “número” em base $b = 256$



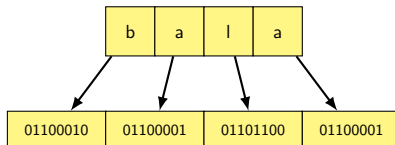
$$x = 'b' \cdot 256^3 + 'a' \cdot 256^2 + 'l' \cdot 256^1 + 'a' \cdot 256^0 = 1650551905$$

Mas x poderia ser muito grande e estourar um `int`...

Método da divisão para Strings

Como podemos obter um número x que representa a chave “bala”?

- Vamos considerar que “bala” é um “número” em base $b = 256$



$$x = \text{'b'} \cdot 256^3 + \text{'a'} \cdot 256^2 + \text{'l'} \cdot 256^1 + \text{'a'} \cdot 256^0 = 1650551905$$

Mas x poderia ser muito grande e estourar um `int`...

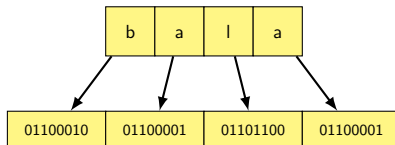
Podemos **reescrever** x como (Método de Horner)

$$x = (((\text{'b'}) \cdot 256 + \text{'a'}) \cdot 256 + \text{'l'}) \cdot 256 + \text{'a'}$$

Método da divisão para Strings

Como podemos obter um número x que representa a chave “bala”?

- Vamos considerar que “bala” é um “número” em base $b = 256$



$$x = \text{'b'} \cdot 256^3 + \text{'a'} \cdot 256^2 + \text{'l'} \cdot 256^1 + \text{'a'} \cdot 256^0 = 1650551905$$

Mas x poderia ser muito grande e estourar um `int`...

Podemos **reescrever** x como (Método de Horner)

$$x = (((\text{'b'}) \cdot 256 + \text{'a'}) \cdot 256 + \text{'l'}) \cdot 256 + \text{'a'}$$

Ao invés de calcular $x \bmod M$, calculamos

$$(((\text{'b'} \bmod M) \cdot 256 + \text{'a'} \bmod M) \cdot 256 + \text{'l'} \bmod M) \cdot 256 + \text{'a'} \bmod M$$

Função de hashing modular

Método da multiplicação:

- posição calculada **não** depende de M (pode ser $M = 1024$)

$$h(\mathbf{x}) = \lfloor M \cdot ([A \cdot \mathbf{x}] \bmod 1) \rfloor$$

Função de hashing modular

Método da multiplicação:

- posição calculada **não** depende de M (pode ser $M = 1024$)

$$h(x) = \lfloor M \cdot ([A \cdot x] \bmod 1) \rfloor$$

- 1 multiplicamos x por um valor real A e obtemos a parte fracionária

Função de hashing modular

Método da multiplicação:

- posição calculada **não** depende de M (pode ser $M = 1024$)

$$h(x) = \lfloor M \cdot ([A \cdot x] \bmod 1) \rfloor$$

- 1 multiplicamos x por um valor real A e obtemos a parte fracionária
- 2 escolhemos A conveniente, por exemplo $A = (\sqrt{5} - 1)/2$

O uso da **razão áurea** como valor de A é sugestão de Knuth: [Fibonacci hashing](#)

Função de hashing modular

Método da multiplicação:

- posição calculada **não** depende de M (pode ser $M = 1024$)

$$h(x) = \lfloor M \cdot ([A \cdot x] \bmod 1) \rfloor$$

- 1 multiplicamos x por um valor real A e obtemos a parte fracionária
- 2 escolhemos A conveniente, por exemplo $A = (\sqrt{5} - 1)/2$

Exemplo:

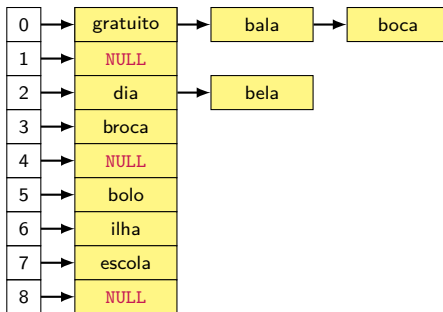
$$\begin{aligned} h(\text{'bala'}) &= \lfloor 1024 \cdot [((\sqrt{5} - 1)/2 \cdot 1650551905) \bmod 1] \rfloor \\ &= \lfloor 1024 \cdot [1020097177,4858876 \bmod 1] \rfloor \\ &= \lfloor 1024 \cdot 0,4858876 \rfloor \\ &= \lfloor 497,5489024 \rfloor = 497 \end{aligned}$$

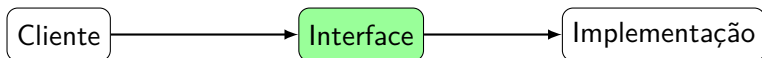
- 1 Introdução
- 2 Função de hashing
- 3 Hashing com encadeamento**
- 4 Hashing com endereçamento aberto
- 5 Referências

Hashing com encadeamento

Resolução de colisões por encadeamento:

- Inserção no início: $O(1)$
- Busca: tempo esperado $O(n/M)$

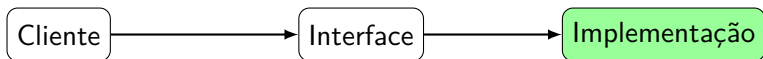




hash.h

```
1  #ifndef HASH_H
2  #define HASH_H
3
4  #include "lista.h"
5
6  //Dados
7  typedef struct {
8      No **vetor;
9      int M; //tamanho do vetor
10 } Hash;
11
12 //Funções
13 Hash* criar_hash(int M);
14 void destruir_hash(Hash **p);
15
16 void inserir_hash(Hash *p, char *chave, int dado);
17 void remover_hash(Hash *p, char *chave);
18
19 int buscar_hash(Hash *p, char *chave); //retorna -1 se nao encontrou
20 void imprimir_hash(Hash *p);
21
22
23 #endif
```

Hashing com encadeamento - Criar e destruir

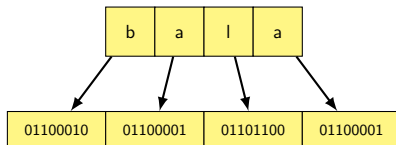


hash.c

```
6 Hash* criar_hash(int M){
7     Hash *p = (Hash*) malloc(sizeof(Hash));
8     p->vetor = (No**) malloc(M * sizeof(No**));
9     int i;
10    for(i=0; i<M; i++)
11        p->vetor[i] = criar_lista();
12    p->M = M;
13    return p;
14 }
```

```
16 void destruir_hash(Hash **p){
17     free((*p)->vetor);
18     free(*p);
19     *p = NULL;
20 }
```

Função de hashing



$$(((\text{'b' mod } M) \cdot 256 + \text{'a' mod } M) \cdot 256 + \text{'l' mod } M) \cdot 256 + \text{'a' mod } M$$

hash.c

```
22 int hashing(char *chave, int M) {  
23     int i, n = 0, len = strlen(chave);  
24     for (i = 0; i < len; i++)  
25         n = (n * 256 + chave[i]) % M;  
26     return n;  
27 }
```

Hashing com encadeamento - Inserir, remover e buscar



hash.c

```
29 void inserir_hash(Hash *p, char *chave, int dado) {  
30     int n = hashing(chave, p->M);  
31     p->vetor[n] = inserir_lista(p->vetor[n], chave, dado);  
32 }
```


Hashing com encadeamento - Inserir, remover e buscar



hash.c

```
29 void inserir_hash(Hash *p, char *chave, int dado) {  
30     int n = hashing(chave, p->M);  
31     p->vetor[n] = inserir_lista(p->vetor[n], chave, dado);  
32 }
```

```
34 void remover_hash(Hash *p, char *chave) {  
35     int n = hashing(chave, p->M);  
36     p->vetor[n] = remover_lista(p->vetor[n], chave);  
37 }
```

Hashing com encadeamento - Inserir, remover e buscar



hash.c

```
29 void inserir_hash(Hash *p, char *chave, int dado) {
30     int n = hashing(chave, p->M);
31     p->vetor[n] = inserir_lista(p->vetor[n], chave, dado);
32 }
```

```
34 void remover_hash(Hash *p, char *chave) {
35     int n = hashing(chave, p->M);
36     p->vetor[n] = remover_lista(p->vetor[n], chave);
37 }
```

```
39 int buscar_hash(Hash *p, char *chave){//retorna -1 se nao encontrou
40     int n = hashing(chave, p->M);
41     return buscar_valor(p->vetor[n], chave);//retorna valor (chave, valor)
42 }
```



exemplo1.c

```
1  #include <stdio.h>
2  #include <string.h>
3  #include "hash.h"
4
5  int main(){
6      char palavras[9][10] = {"broca", "boca", "bolo", "bela", "bala",
7                              "dia", "escola", "gratuito", "ilha"};
8      int i;
9      Hash *H = criar_hash(9);
10     for(i = 0; i < 9; i++)
11         inserir_hash(H, palavras[i], i);
12     imprimir_hash(H);
13     destruir_hash(&H);
14     return 0;
15 }
```

Makefile

Vamos usar o [Makefile](#) para compilar:

```
1 LIB = hash.o lista.o
```

```
1 exemplo1: exemplo1.c $(LIB)
2 gcc $^ -o $@
```

Vamos executar:

```
1 $ ./exemplo1
2 0: (gratuito, 7) -> (boca, 1)
3 1: (bala, 4)
4 2: (dia, 5) -> (bela, 3)
5 3: (broca, 0)
6 4:
7 5: (bolo, 2)
8 6: (ilha, 8)
9 7: (escola, 6)
10 8:
```

- 1 Introdução
- 2 Função de hashing
- 3 Hashing com encadeamento
- 4 Hashing com endereçamento aberto**
- 5 Referências

Endereçamento aberto

Existe uma alternativa para a **implementação** de **tabela de espalhamento**

Endereçamento aberto:

- os dados são guardados no próprio vetor
- **colisões** são colocadas em **posições livres** da tabela

Endereçamento aberto

Existe uma alternativa para a **implementação** de **tabela de espalhamento**

Endereçamento aberto:

- os dados são guardados no próprio vetor
- **colisões** são colocadas em **posições livres** da tabela

Características:

- ① evita percorrer usando ponteiros e alocação e deslocação de memória (**malloc** e **free**)

Endereçamento aberto

Existe uma alternativa para a **implementação** de **tabela de espalhamento**

Endereçamento aberto:

- os dados são guardados no próprio vetor
- **colisões** são colocadas em **posições livres** da tabela

Características:

- ① evita percorrer usando ponteiros e alocação e deslocação de memória (**malloc** e **free**)
- ② se a tabela encher, deve recriar uma tabela maior (e mudar a função de hashing)

Endereçamento aberto

Existe uma alternativa para a **implementação** de **tabela de espalhamento**

Endereçamento aberto:

- os dados são guardados no próprio vetor
- **colisões** são colocadas em **posições livres** da tabela

Características:

- ❶ evita percorrer usando ponteiros e alocação e deslocação de memória (**malloc** e **free**)
- ❷ se a tabela encher, deve recriar uma tabela maior (e mudar a função de hashing)
- ❸ remoção é mais complicada

Endereçamento aberto com sondagem linear

broca

boca

bolo

bela

bala

dia

escola

gratuito

ilha

0	
1	
2	
3	
4	
5	
6	
7	
8	

Inserindo:

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca

bolo

bela

bala

dia

escola

gratuito

ilha

0	
1	
2	
3	
4	
5	
6	
7	
8	

Inserindo:

- **procuramos** posição

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca

bolo

bela

bala

dia

escola

gratuito

ilha

0	
1	
2	
3	
4	
5	
6	
7	
8	



Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca

bolo

bela

bala

dia

escola

gratuito

ilha

0	
1	
2	
3	broca
4	
5	
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo

bela

bala

dia

escola

gratuito

ilha

0	
1	
2	
3	broca
4	
5	
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo

bela


bala

dia

escola

gratuito

ilha



0	
1	
2	
3	broca
4	
5	
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo

bela

bala

dia

escola

gratuito

ilha

0	boca
1	
2	
3	broca
4	
5	
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela

bala

dia

escola

gratuito

ilha

0	boca
1	
2	
3	broca
4	
5	
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela

bala

dia

escola

gratuito

ilha

0	boca
1	
2	
3	broca
4	
5	
6	
7	
8	



Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela

bala

dia

escola

gratuito

ilha

0	boca
1	
2	
3	broca
4	
5	bolo
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala

dia

escola

gratuito

ilha

0	boca
1	
2	
3	broca
4	
5	bolo
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos

Endereçamento aberto com sondagem linear

broca \rightsquigarrow $h(\text{"broca"}) = 3$

boca \rightsquigarrow $h(\text{"boca"}) = 0$

bolo \rightsquigarrow $h(\text{"bolo"}) = 5$

bela \rightsquigarrow $h(\text{"bela"}) = 2$


bala

dia

escola

gratuito

ilha



0	boca
1	
2	
3	broca
4	
5	bolo
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala

dia

escola

gratuito

ilha

0	boca
1	
2	bela
3	broca
4	
5	bolo
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$


bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia

escola

gratuito

ilha



0	boca
1	
2	bela
3	broca
4	
5	bolo
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia

escola

gratuito

ilha



0	boca
1	
2	bela
3	broca
4	
5	bolo
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** *M*)

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia

escola

gratuito

ilha

0	boca
1	bala
2	bela
3	broca
4	
5	bolo
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia

escola

gratuito

ilha

0	boca
1	bala
2	bela
3	broca
4	
5	bolo
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola

gratuito

ilha

0	boca
1	bala
2	bela
3	broca
4	
5	bolo
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola

gratuito

ilha

0	boca
1	bala
2	bela
3	broca
4	
5	bolo
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$


bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola

gratuito

ilha



0	boca
1	bala
2	bela
3	broca
4	
5	bolo
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$


bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola

gratuito

ilha



0	boca
1	bala
2	bela
3	broca
4	
5	bolo
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola

gratuito

ilha

0	boca
1	bala
2	bela
3	broca
4	
5	bolo
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola

gratuito

ilha

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca \rightsquigarrow $h(\text{"broca"}) = 3$

boca \rightsquigarrow $h(\text{"boca"}) = 0$

bolo \rightsquigarrow $h(\text{"bolo"}) = 5$

bela \rightsquigarrow $h(\text{"bela"}) = 2$

bala \rightsquigarrow $h(\text{"bala"}) = 0$

dia \rightsquigarrow $h(\text{"dia"}) = 2$

escola \rightsquigarrow $h(\text{"escola"}) = 7$

gratuito

ilha

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	
7	
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola $\rightsquigarrow h(\text{"escola"}) = 7$

gratuito

ilha

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	
7	
8	



Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola $\rightsquigarrow h(\text{"escola"}) = 7$

gratuito

ilha

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	
7	escola
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola $\rightsquigarrow h(\text{"escola"}) = 7$

gratuito $\rightsquigarrow h(\text{"gratuito"}) = 0$

ilha

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	
7	escola
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca \rightsquigarrow $h(\text{"broca"}) = 3$

boca \rightsquigarrow $h(\text{"boca"}) = 0$

bolo \rightsquigarrow $h(\text{"bolo"}) = 5$

bela \rightsquigarrow $h(\text{"bela"}) = 2$


bala \rightsquigarrow $h(\text{"bala"}) = 0$

dia \rightsquigarrow $h(\text{"dia"}) = 2$

escola \rightsquigarrow $h(\text{"escola"}) = 7$

gratuito \rightsquigarrow $h(\text{"gratuito"}) = 0$

ilha



0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	
7	escola
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca \rightsquigarrow $h(\text{"broca"}) = 3$

boca \rightsquigarrow $h(\text{"boca"}) = 0$

bolo \rightsquigarrow $h(\text{"bolo"}) = 5$

bela \rightsquigarrow $h(\text{"bela"}) = 2$


bala \rightsquigarrow $h(\text{"bala"}) = 0$

dia \rightsquigarrow $h(\text{"dia"}) = 2$

escola \rightsquigarrow $h(\text{"escola"}) = 7$

gratuito \rightsquigarrow $h(\text{"gratuito"}) = 0$

ilha



0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	
7	escola
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca \rightsquigarrow $h(\text{"broca"}) = 3$

boca \rightsquigarrow $h(\text{"boca"}) = 0$

bolo \rightsquigarrow $h(\text{"bolo"}) = 5$

bela \rightsquigarrow $h(\text{"bela"}) = 2$


bala \rightsquigarrow $h(\text{"bala"}) = 0$

dia \rightsquigarrow $h(\text{"dia"}) = 2$

escola \rightsquigarrow $h(\text{"escola"}) = 7$

gratuito \rightsquigarrow $h(\text{"gratuito"}) = 0$

ilha



0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	
7	escola
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$


bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola $\rightsquigarrow h(\text{"escola"}) = 7$

gratuito $\rightsquigarrow h(\text{"gratuito"}) = 0$

ilha



0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	
7	escola
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca \rightsquigarrow $h(\text{"broca"}) = 3$

boca \rightsquigarrow $h(\text{"boca"}) = 0$

bolo \rightsquigarrow $h(\text{"bolo"}) = 5$

bela \rightsquigarrow $h(\text{"bela"}) = 2$


bala \rightsquigarrow $h(\text{"bala"}) = 0$

dia \rightsquigarrow $h(\text{"dia"}) = 2$

escola \rightsquigarrow $h(\text{"escola"}) = 7$

gratuito \rightsquigarrow $h(\text{"gratuito"}) = 0$

ilha



0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	
7	escola
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca \rightsquigarrow $h(\text{"broca"}) = 3$

boca \rightsquigarrow $h(\text{"boca"}) = 0$

bolo \rightsquigarrow $h(\text{"bolo"}) = 5$

bela \rightsquigarrow $h(\text{"bela"}) = 2$

bala \rightsquigarrow $h(\text{"bala"}) = 0$

dia \rightsquigarrow $h(\text{"dia"}) = 2$

escola \rightsquigarrow $h(\text{"escola"}) = 7$

gratuito \rightsquigarrow $h(\text{"gratuito"}) = 0$

ilha

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	
7	escola
8	



Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola $\rightsquigarrow h(\text{"escola"}) = 7$

gratuito $\rightsquigarrow h(\text{"gratuito"}) = 0$

ilha

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	
7	escola
8	



Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca \rightsquigarrow $h(\text{"broca"}) = 3$

boca \rightsquigarrow $h(\text{"boca"}) = 0$

bolo \rightsquigarrow $h(\text{"bolo"}) = 5$

bela \rightsquigarrow $h(\text{"bela"}) = 2$

bala \rightsquigarrow $h(\text{"bala"}) = 0$

dia \rightsquigarrow $h(\text{"dia"}) = 2$

escola \rightsquigarrow $h(\text{"escola"}) = 7$

gratuito \rightsquigarrow $h(\text{"gratuito"}) = 0$

ilha

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	gratuito
7	escola
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (**mod** M)

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola $\rightsquigarrow h(\text{"escola"}) = 7$

gratuito $\rightsquigarrow h(\text{"gratuito"}) = 0$

ilha $\rightsquigarrow h(\text{"ilha"}) = 6$

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	gratuito
7	escola
8	

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (mod M)

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola $\rightsquigarrow h(\text{"escola"}) = 7$

gratuito $\rightsquigarrow h(\text{"gratuito"}) = 0$

ilha $\rightsquigarrow h(\text{"ilha"}) = 6$

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	gratuito
7	escola
8	



Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (mod M)

Endereçamento aberto com sondagem linear

broca \rightsquigarrow $h(\text{"broca"}) = 3$

boca \rightsquigarrow $h(\text{"boca"}) = 0$

bolo \rightsquigarrow $h(\text{"bolo"}) = 5$

bela \rightsquigarrow $h(\text{"bela"}) = 2$

bala \rightsquigarrow $h(\text{"bala"}) = 0$

dia \rightsquigarrow $h(\text{"dia"}) = 2$

escola \rightsquigarrow $h(\text{"escola"}) = 7$

gratuito \rightsquigarrow $h(\text{"gratuito"}) = 0$

ilha \rightsquigarrow $h(\text{"ilha"}) = 6$

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	gratuito
7	escola
8	



Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (mod M)

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola $\rightsquigarrow h(\text{"escola"}) = 7$

gratuito $\rightsquigarrow h(\text{"gratuito"}) = 0$

ilha $\rightsquigarrow h(\text{"ilha"}) = 6$

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	gratuito
7	escola
8	



Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (mod M)

Endereçamento aberto com sondagem linear

broca $\rightsquigarrow h(\text{"broca"}) = 3$

boca $\rightsquigarrow h(\text{"boca"}) = 0$

bolo $\rightsquigarrow h(\text{"bolo"}) = 5$

bela $\rightsquigarrow h(\text{"bela"}) = 2$

bala $\rightsquigarrow h(\text{"bala"}) = 0$

dia $\rightsquigarrow h(\text{"dia"}) = 2$

escola $\rightsquigarrow h(\text{"escola"}) = 7$

gratuito $\rightsquigarrow h(\text{"gratuito"}) = 0$

ilha $\rightsquigarrow h(\text{"ilha"}) = 6$

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	gratuito
7	escola
8	ilha

Inserindo:

- **procuramos** posição
- se **houver** espaço, guardamos
- se **não houver** espaço, procuramos a próxima posição livre (mod M)

Busca em endereçamento aberto

Como fazer uma busca com endereçamento aberto?

- 1 Calcule a **função de hashing**
- 2 Percorra a tabela **em sequência** procurando pela chave
- 3 Se **encontrar a chave**, devolva o item correspondente
- 4 Se **encontrar um espaço vazio**, devolva **NULL**

gratuito $\rightsquigarrow h(\text{"gratuito"}) = 0$

ilha $\rightsquigarrow h(\text{"ilha"}) = 6$

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	gratuito
7	escola
8	

Busca em endereçamento aberto

Como fazer uma busca com endereçamento aberto?

- 1 Calcule a **função de hashing**
- 2 Percorra a tabela **em sequência** procurando pela chave
- 3 Se **encontrar a chave**, devolva o item correspondente
- 4 Se **encontrar um espaço vazio**, devolva **NULL**

gratuito $\rightsquigarrow h(\text{"gratuito"}) = 0$

ilha $\rightsquigarrow h(\text{"ilha"}) = 6$

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	gratuito
7	escola
8	

O que é um espaço vazio em um vetor?

- Se for um vetor de ponteiros, pode ser **NULL**

Busca em endereçamento aberto

Como fazer uma busca com endereçamento aberto?

- 1 Calcule a **função de hashing**
- 2 Percorra a tabela **em sequência** procurando pela chave
- 3 Se **encontrar a chave**, devolva o item correspondente
- 4 Se **encontrar um espaço vazio**, devolva **NULL**

gratuito $\rightsquigarrow h(\text{"gratuito"}) = 0$

ilha $\rightsquigarrow h(\text{"ilha"}) = 6$

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	gratuito
7	escola
8	

O que é um espaço vazio em um vetor?

- Se for um vetor de ponteiros, pode ser **NULL**
- Se não for, precisa ser um elemento **dummy**
 - um valor que **nunca** será usado, ou ter um campo indicando que é **dummy**

Remoção em endereçamento aberto

Como fazer a remoção com endereçamento aberto?

- Não podemos **apenas remover** os elementos da tabela
 - Por quê?

dia $\rightsquigarrow h(\text{"dia"}) = 2$

gratuito $\rightsquigarrow h(\text{"gratuito"}) = 0$

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	gratuito
7	escola
8	

Remoção em endereçamento aberto

Como fazer a remoção com endereçamento aberto?

- Não podemos **apenas remover** os elementos da tabela
 - Por quê? Quebraria a busca...

dia $\rightsquigarrow h(\text{"dia"}) = 2$

gratuito $\rightsquigarrow h(\text{"gratuito"}) = 0$

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	gratuito
7	escola
8	

Remoção em endereçamento aberto

Como fazer a remoção com endereçamento aberto?

- Não podemos **apenas remover** os elementos da tabela
 - Por quê? Quebraria a busca...

dia $\rightsquigarrow h(\text{"dia"}) = 2$

gratuito $\rightsquigarrow h(\text{"gratuito"}) = 0$

Algumas opções:

- 1 rehash dos elementos seguintes do bloco
 - **removemos** os elementos até a próxima posição vazia
 - recalculamos o hash **de cada um** e reinserimos na tabela
 - pode ser custoso e difícil de implementar

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	gratuito
7	escola
8	

Remoção em endereçamento aberto

Como fazer a remoção com endereçamento aberto?

- Não podemos **apenas remover** os elementos da tabela
 - Por quê? Quebraria a busca...

dia $\rightsquigarrow h(\text{"dia"}) = 2$

gratuito $\rightsquigarrow h(\text{"gratuito"}) = 0$

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	gratuito
7	escola
8	

Algumas opções:

- 1 rehash dos elementos seguintes do bloco
 - **removemos** os elementos até a próxima posição vazia
 - **recalculamos** o hash **de cada um** e reinserimos na tabela
 - pode ser custoso e difícil de implementar
- 2 trocamos por um valor **dummy**
 - valor indica que o item **foi removido**
 - mas **não pode** ser o mesmo que indica espaço vazio

Remoção em endereçamento aberto

Como fazer a remoção com endereçamento aberto?

- Não podemos **apenas remover** os elementos da tabela
 - Por quê? Quebraria a busca...

dia $\rightsquigarrow h(\text{"dia"}) = 2$

gratuito $\rightsquigarrow h(\text{"gratuito"}) = 0$

0	boca
1	bala
2	bela
3	broca
4	dia
5	bolo
6	gratuito
7	escola
8	

Algumas opções:

- 1 rehash dos elementos seguintes do bloco
 - **removemos** os elementos até a próxima posição vazia
 - **recalculamos** o hash **de cada um** e reinserimos na tabela
 - pode ser custoso e difícil de implementar
- 2 trocamos por um valor **dummy**
 - valor indica que o item **foi removido**
 - mas **não pode** ser o mesmo que indica espaço vazio
- 3 marcamos o item como removido
 - usamos um campo adicional

Hashing duplo

Hashing duplo:

- Estratégia mais geral para lidar com **conflitos**
- Ao invés de saltarmos sempre de 1, saltamos de $h_2(x)$
- Onde h_2 é uma segunda função de hashing

$$h_2(x) = (x \bmod (M - 1)) + 1$$

0	boca
1	bala
2	bela
3	broca
4	
5	bolo
6	
7	
8	

Hashing duplo

Hashing duplo:

- Estratégia mais geral para lidar com **conflitos**
- Ao invés de saltarmos sempre de 1, saltamos de $h_2(x)$
- Onde h_2 é uma segunda função de hashing

$$h_2(x) = (x \bmod (M - 1)) + 1$$

Alguns cuidados:

- $h_2(k)$ **nunca** pode ser zero

0	boca
1	bala
2	bela
3	broca
4	
5	bolo
6	
7	
8	

Hashing duplo

Hashing duplo:

- Estratégia mais geral para lidar com **conflitos**
- Ao invés de saltarmos sempre de 1, saltamos de $h_2(x)$
- Onde h_2 é uma segunda função de hashing

$$h_2(x) = (x \bmod (M - 1)) + 1$$

Alguns cuidados:

- $h_2(k)$ **nunca** pode ser zero
- $h_2(k)$ e M devem ser **primos entre si** (coprimos): o que garante a soma do índice atual com o deslocamento dê um número diferente até **verificar** todos as M posições:

$$h(x, i) = (h_1(x) + i \cdot h_2(x)) \bmod M$$

0	boca
1	bala
2	bela
3	broca
4	
5	bolo
6	
7	
8	

Hashing duplo

Hashing duplo:

- Estratégia mais geral para lidar com **conflitos**
- Ao invés de saltarmos sempre de 1, saltamos de $h_2(x)$
- Onde h_2 é uma segunda função de hashing

$$h_2(x) = (x \bmod (M - 1)) + 1$$

0	boca
1	bala
2	bela
3	broca
4	
5	bolo
6	
7	
8	

Alguns cuidados:

- $h_2(k)$ **nunca** pode ser zero
- $h_2(k)$ e M devem ser **primos entre si** (coprimos): o que garante a soma do índice atual com o deslocamento dê um número diferente até **verificar** todos as M posições:

$$h(x, i) = (h_1(x) + i \cdot h_2(x)) \bmod M$$

Exemplo:

- Escolha M como um número **primo** e faça que $h_2(k) < M$

Sondagem linear e Hashing duplo¹

Sondagem linear:

n/M	1/2	2/3	3/4	9/10
com sucesso	1.5	2.0	3.0	5.5
sem sucesso	2.5	5.0	8.5	55.5

Tabela: número de acessos médio por busca

0	boca
1	bala
2	bela
3	broca
4	
5	bolo
6	
7	
8	

¹Baseado em Sedgwick, R. Algorithms in C, third edition, Addison-Wesley. 1998.

Sondagem linear e Hashing duplo¹

Sondagem linear:

n/M	1/2	2/3	3/4	9/10
com sucesso	1.5	2.0	3.0	5.5
sem sucesso	2.5	5.0	8.5	55.5

Tabela: número de acessos médio por busca

Hashing duplo:

n/M	1/2	2/3	3/4	9/10
com sucesso	1.4	1.6	1.8	2.6
sem sucesso	1.5	2.0	3.0	5.5

Tabela: número de acessos médio por busca

0	boca
1	bala
2	bela
3	broca
4	
5	bolo
6	
7	
8	

¹Baseado em Sedgewick, R. Algorithms in C, third edition, Addison-Wesley. 1998.

Sondagem linear e Hashing duplo¹

Sondagem linear:

n/M	1/2	2/3	3/4	9/10
com sucesso	1.5	2.0	3.0	5.5
sem sucesso	2.5	5.0	8.5	55.5

Tabela: número de acessos médio por busca

Hashing duplo:

n/M	1/2	2/3	3/4	9/10
com sucesso	1.4	1.6	1.8	2.6
sem sucesso	1.5	2.0	3.0	5.5

Tabela: número de acessos médio por busca

0	boca
1	bala
2	bela
3	broca
4	
5	bolo
6	
7	
8	

De qualquer forma, é importante não deixar a tabela encher muito:

- Você pode redimensionar a tabela dinamicamente (precisa *rehash*)

¹Baseado em Sedgewick, R. Algorithms in C, third edition, Addison-Wesley. 1998.

Hashing é uma **boa** estrutura de dados para

- inserir, remover e buscar dados pela sua chave **rapidamente**
- com uma boa **função de hashing**, essas operações tem tempo $O(1)$
- mas **não é boa** se quisermos fazer operação relacionadas a ordem das chaves

Hashing é uma **boa** estrutura de dados para

- inserir, remover e buscar dados pela sua chave **rapidamente**
- com uma boa **função de hashing**, essas operações tem tempo $O(1)$
- mas **não é boa** se quisermos fazer operação relacionadas a ordem das chaves

Escolhendo a implementação:

- 1 Encadeamento separado é mais **fácil** de implementar

Hashing é uma **boa** estrutura de dados para

- inserir, remover e buscar dados pela sua chave **rapidamente**
- com uma boa **função de hashing**, essas operações tem tempo $O(1)$
- mas **não é boa** se quisermos fazer operação relacionadas a ordem das chaves

Escolhendo a implementação:

- 1 Encadeamento separado é mais **fácil** de implementar
 - Usa memória a mais para os ponteiros

Hashing é uma **boa** estrutura de dados para

- inserir, remover e buscar dados pela sua chave **rapidamente**
- com uma boa **função de hashing**, essas operações tem tempo $O(1)$
- mas **não é boa** se quisermos fazer operação relacionadas a ordem das chaves

Escolhendo a implementação:

- 1 Encadeamento separado é mais **fácil** de implementar
 - Usa memória a mais para os ponteiros
- 2 Sondagem linear é o mais **rápido** se a tabela for esparsa

Hashing é uma **boa** estrutura de dados para

- inserir, remover e buscar dados pela sua chave **rapidamente**
- com uma boa **função de hashing**, essas operações tem tempo $O(1)$
- mas **não é boa** se quisermos fazer operação relacionadas a ordem das chaves

Escolhendo a implementação:

- 1 Encadeamento separado é mais **fácil** de implementar
 - Usa memória a mais para os ponteiros
- 2 Sondagem linear é o mais **rápido** se a tabela for esparsa
- 3 Hashing duplo usa **melhor** a **memória**

Hashing é uma **boa** estrutura de dados para

- inserir, remover e buscar dados pela sua chave **rapidamente**
- com uma boa **função de hashing**, essas operações tem tempo $O(1)$
- mas **não é boa** se quisermos fazer operação relacionadas a ordem das chaves

Escolhendo a implementação:

- 1 Encadeamento separado é mais **fácil** de implementar
 - Usa memória a mais para os ponteiros
- 2 Sondagem linear é o mais **rápido** se a tabela for esparsa
- 3 Hashing duplo usa **melhor** a **memória**
 - mas gasta mais tempo para computar a segunda função de hash

Dúvidas?

- 1 Introdução
- 2 Função de hashing
- 3 Hashing com encadeamento
- 4 Hashing com endereçamento aberto
- 5 Referências**

- ① Materiais adaptados dos slides do Prof. Rafael C. S. Schouery, da Universidade Estadual de Campinas.
- ② R. Sedgewick, "*Algorithms in C - Parts 1-4 - Third Edition*" (Capítulo 14)