Tabelas de Dispersão

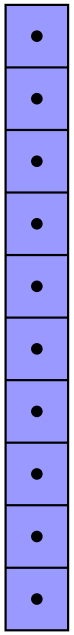
HASH

Prof. Kennedy Lopes

26 de Agosto de 2021

Motivação

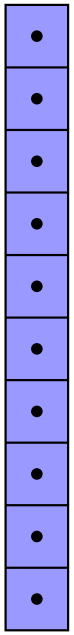
Acesso a um dado com um vetor é muito eficiente:



Motivação

Acesso a um dado com um vetor é muito eficiente:

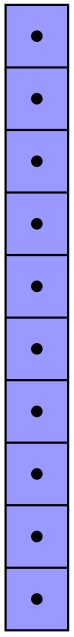
```
Info *v[10];
```



Motivação

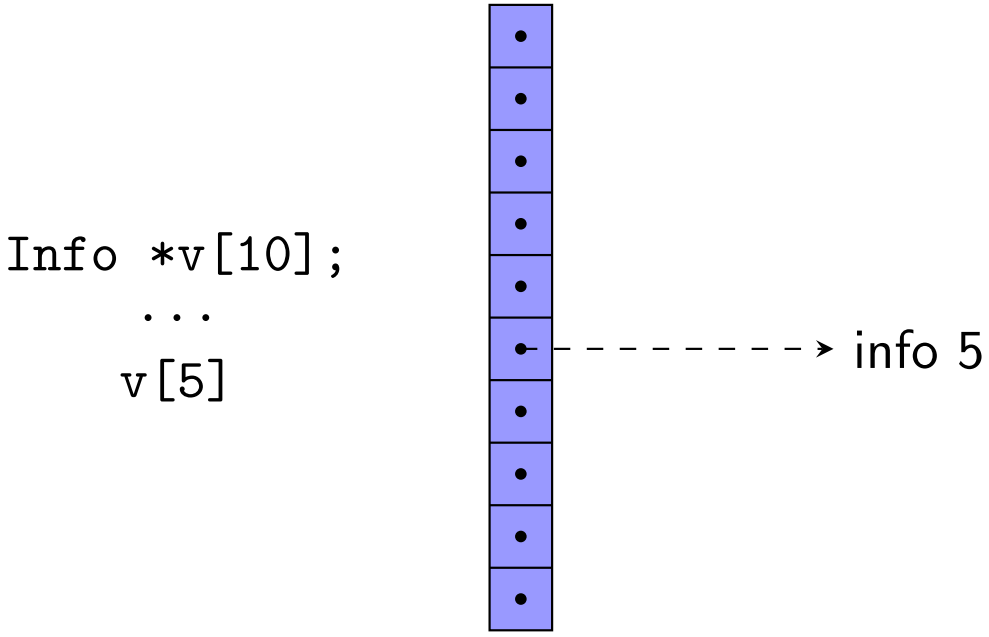
Acesso a um dado com um vetor é muito eficiente:

```
Info *v[10];
     ...
```



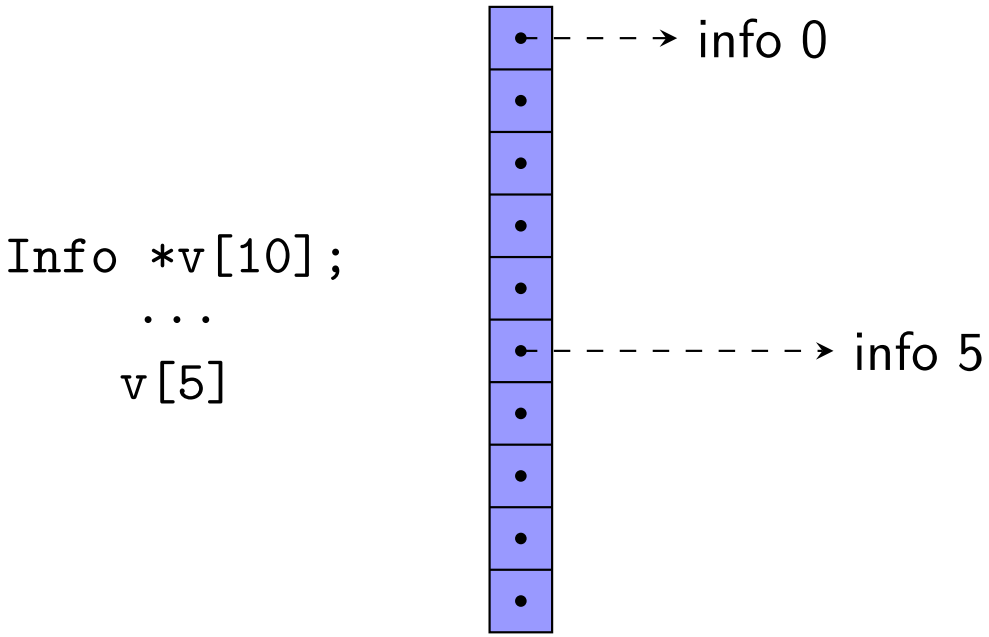
Motivação

Acesso a um dado com um vetor é muito eficiente:



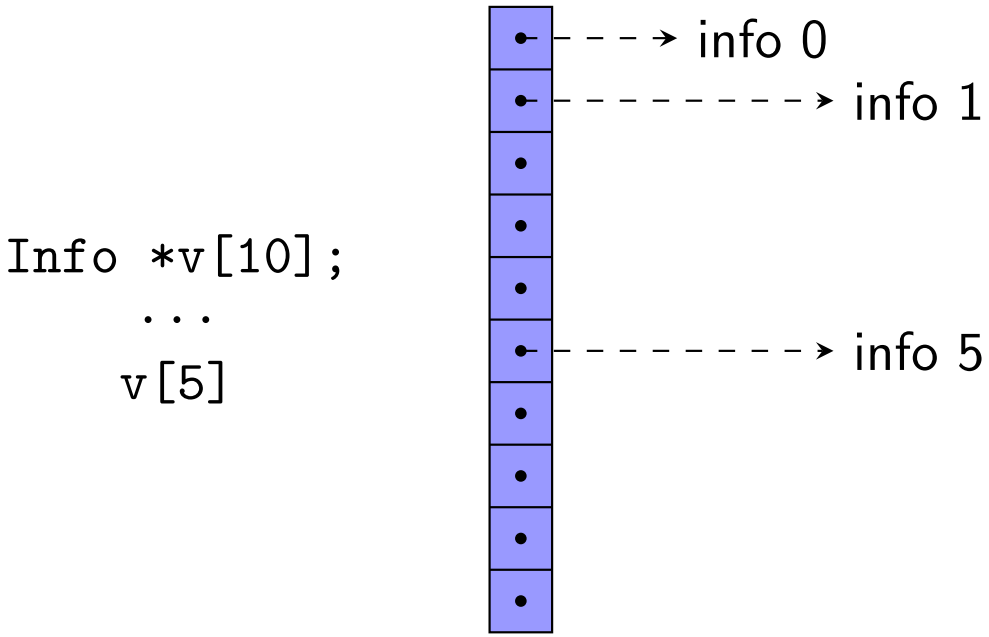
Motivação

Acesso a um dado com um vetor é muito eficiente:



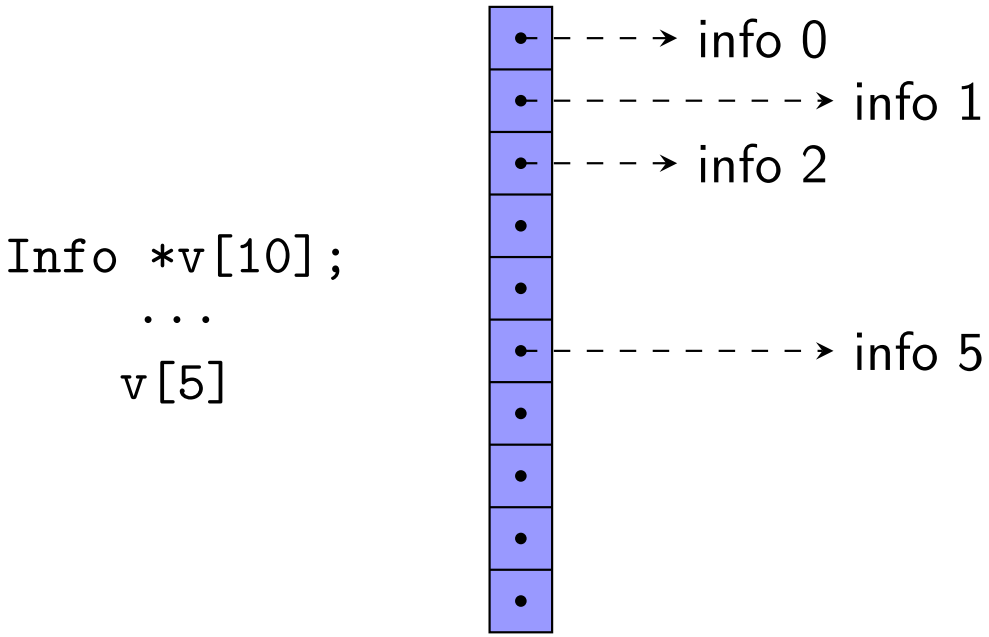
Motivação

Acesso a um dado com um vetor é muito eficiente:



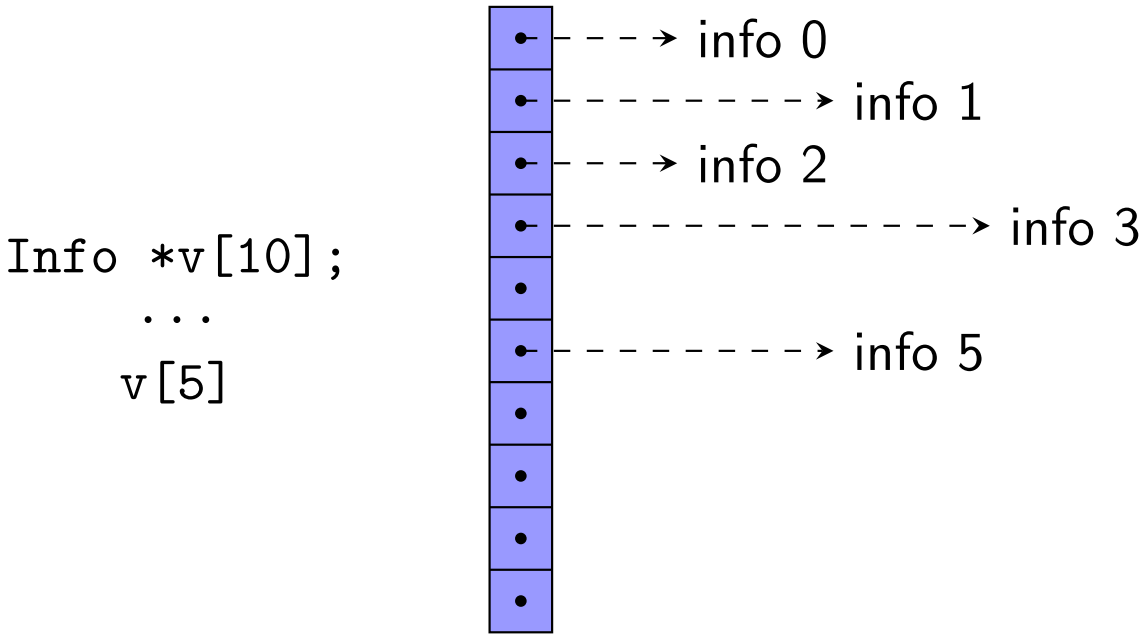
Motivação

Acesso a um dado com um vetor é muito eficiente:



Motivação

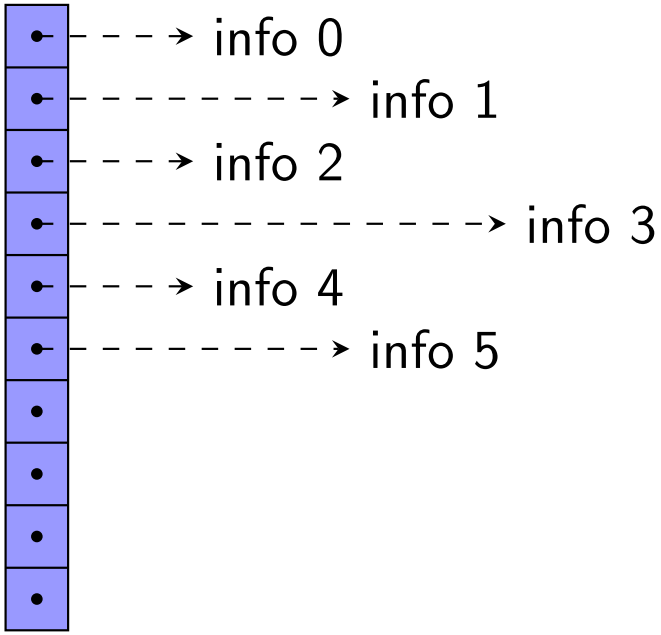
Acesso a um dado com um vetor é muito eficiente:



Motivação

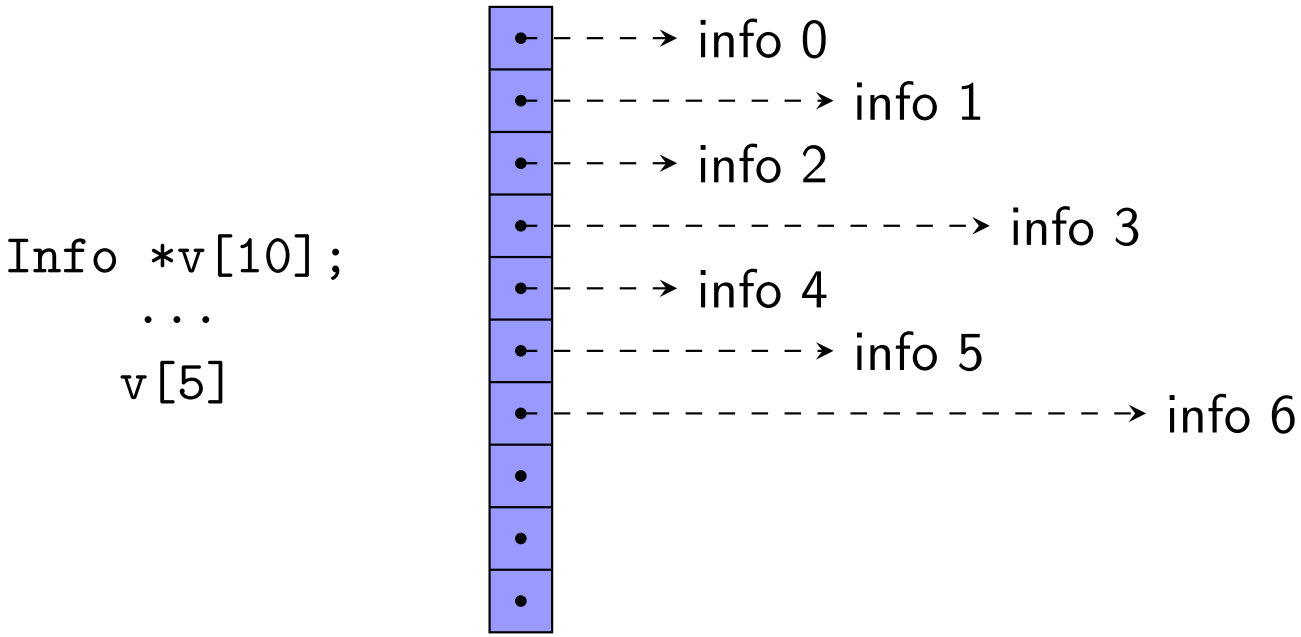
Acesso a um dado com um vetor é muito eficiente:

```
Info *v[10];
   ...
  v[5]
```



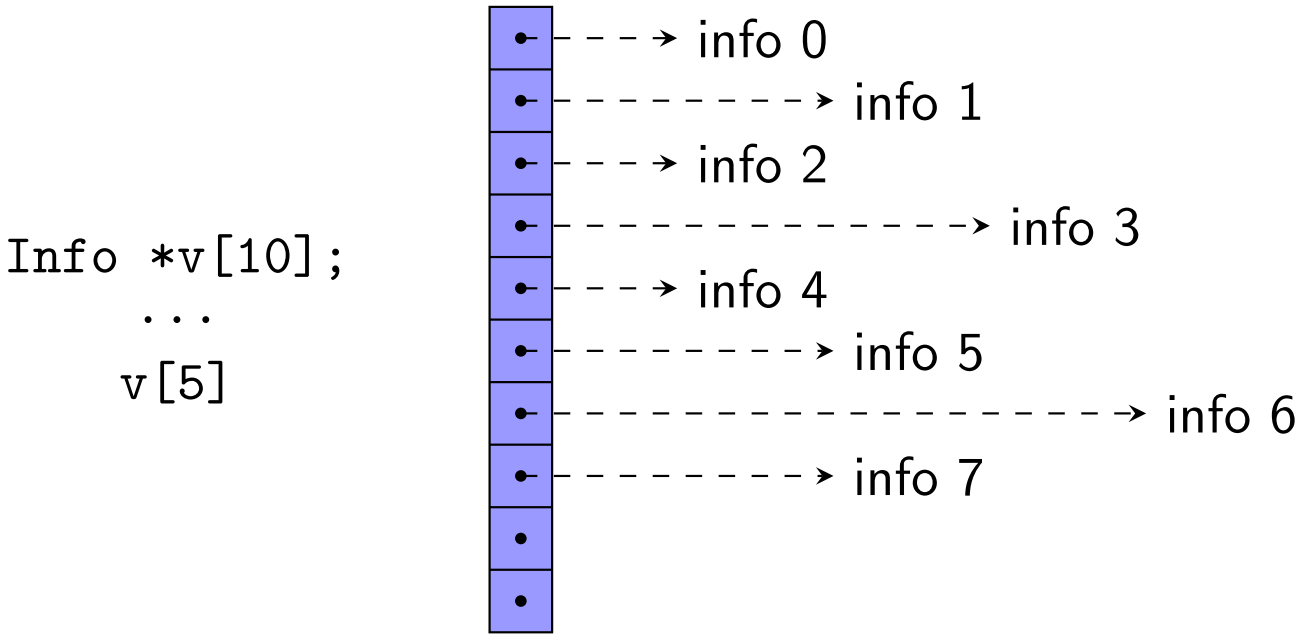
Motivação

Acesso a um dado com um vetor é muito eficiente:



Motivação

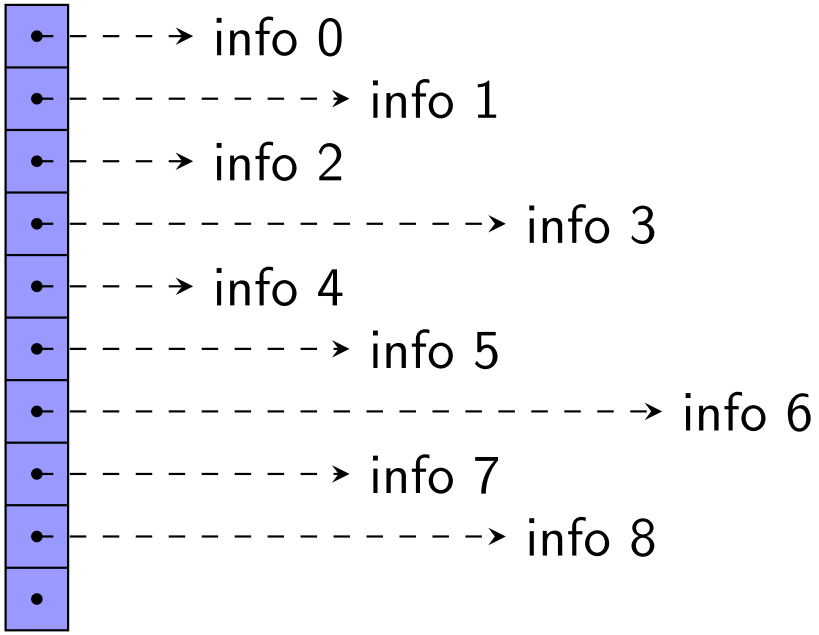
Acesso a um dado com um vetor é muito eficiente:



Motivação

Acesso a um dado com um vetor é muito eficiente:

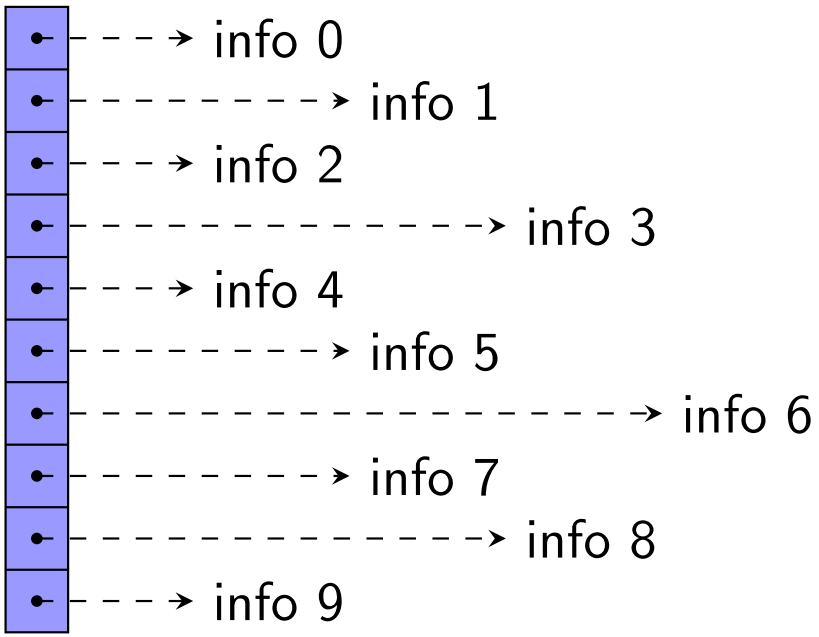
```
Info *v[10];
   ...
  v[5]
```



Motivação

Acesso a um dado com um vetor é muito eficiente:

```
Info *v[10];
...
v[5]
```



Motivação

Geralmente as chaves numéricas são grandes.

Aluno
int mat;

Motivação

Geralmente as chaves numéricas são grandes.

Aluno
int mat;
char nome[81];

Motivação

Geralmente as chaves numéricas são grandes.

Aluno
int mat;
char nome[81];
char email[61]

Motivação

Geralmente as chaves numéricas são grandes.

Aluno
int mat;
char nome[81];
char email[61]

- Uma busca por nome ou matrícula em um vetor pode ser muito dispendioso.

Motivação

Geralmente as chaves numéricas são grandes.

Aluno
int mat;
char nome[81];
char email[61]

- Uma busca por nome ou matrícula em um vetor pode ser muito dispendioso.
- Muito armazenamento devem ser previamente (e ociosamente) disponibilizados.

Motivação

Ideia de caixa postal:

Motivação

Ideia de caixa postal:

- A primeira letra define a caixa onde a informação é adicionada;

Motivação

Ideia de caixa postal:

- A primeira letra define a caixa onde a informação é adicionada;
- O endereçamento é realizado por uma função *Hash*;

Motivação

Ideia de caixa postal:

- A primeira letra define a caixa onde a informação é adicionada;
- O endereçamento é realizado por uma função *Hash*;
- Função *Hash*:

$$h : [0, N] \rightarrow [0, m]$$
$$x \rightarrow h(x)$$

Correios

Conceito de *Hash*

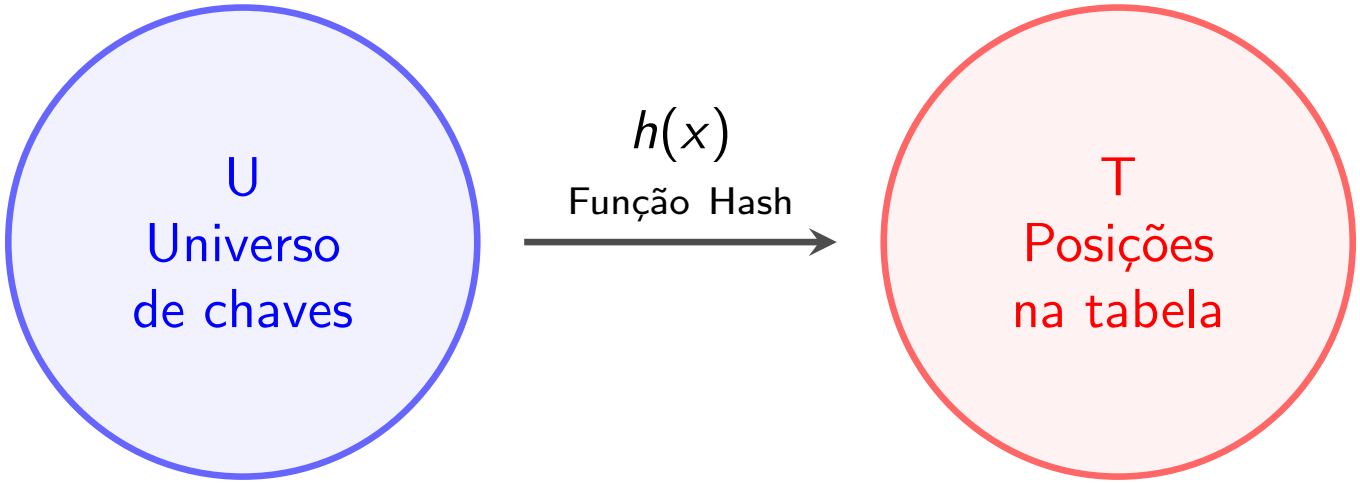
Geralmente os métodos de busca realizam a comparação direta do valor da chave com o termo procurado. Com função *hash* outro termo é buscado devido a uma ordenação das entradas.

- **Conceito:** Os registros armazenados em uma tabela são endereçados a partir de uma transformação aritmética sobre a chave de pesquisa;
- **Objetivo**: Ter eficiência $O(1)$ nas buscas, inserções e remoções. Para isso a escolha do conjunto **[0, m]** (imagem da função) deve ser escolhido adequadamente.

Conceito de *Hash*

Conceito de *Hash*

Conceito de *Hash*



Conceito de *Hash*

O método de pesquisa conhecido como *Hash* (tabela de dispersão) é constituído de duas etapas principais:

- A chave de pesquisa é transformada em um endereço de uma tabela.
- Caso duas chaves sejam mapeadas na mesma posição, a **colisão**:
  - O dado pode ser rejeitado;
  - Uma estrutura auxiliar acomoda as duas chaves na mesma linha da tabela.

Hash table

15	∅
14	∅
13	∅
12	∅
11	∅
10	∅
9	∅
8	∅
7	∅
6	∅
5	∅
4	∅
3	∅
2	∅
1	∅
0	∅

$$h(k) = mod(k, 16)$$

$$k \in \{61, 90, 21, 32\}$$

Conceito de *Hash*

O método de pesquisa conhecido como *Hash* (tabela de dispersão) é constituído de duas etapas principais:

- A chave de pesquisa é transformada em um endereço de uma tabela.
- Caso duas chaves sejam mapeadas na mesma posição, a **colisão**:
  - O dado pode ser rejeitado;
  - Uma estrutura auxiliar acomoda as duas chaves na mesma linha da tabela.

Hash table

15	∅
14	∅
13	**61**
12	∅
11	∅
10	∅
9	∅
8	∅
7	∅
6	∅
5	∅
4	∅
3	∅
2	∅
1	∅
0	∅

$$h(k) = mod(k, 16)$$

$$k \in \{61, 90, 21, 32\}$$

$$h(61) = mod(61, 16) = 13$$

Conceito de *Hash*

O método de pesquisa conhecido como *Hash* (tabela de dispersão) é constituído de duas etapas principais:

- A chave de pesquisa é transformada em um endereço de uma tabela.
- Caso duas chaves sejam mapeadas na mesma posição, a **colisão**:
  - O dado pode ser rejeitado;
  - Uma estrutura auxiliar acomoda as duas chaves na mesma linha da tabela.

Hash table

15	∅
14	∅
13	**61**
12	∅
11	∅
10	**90**
9	∅
8	∅
7	∅
6	∅
5	∅
4	∅
3	∅
2	∅
1	∅
0	∅

$$h(k) = mod(k, 16)$$

$$k \in \{61, 90, 21, 32\}$$

$$h(90) = mod(90, 16) = 10$$

Conceito de *Hash*

O método de pesquisa conhecido como *Hash* (tabela de dispersão) é constituído de duas etapas principais:

- A chave de pesquisa é transformada em um endereço de uma tabela.
- Caso duas chaves sejam mapeadas na mesma posição, a **colisão**:
  - O dado pode ser rejeitado;
  - Uma estrutura auxiliar acomoda as duas chaves na mesma linha da tabela.

Hash table

15	∅
14	∅
13	**61**
12	∅
11	∅
10	**90**
9	∅
8	∅
7	∅
6	∅
5	**21**
4	∅
3	∅
2	∅
1	∅
0	∅

$$h(k) = mod(k, 16)$$

$$k \in \{61, 90, 21, 32\}$$

$$h(21) = mod(21, 16) = 5$$

Conceito de *Hash*

O método de pesquisa conhecido como *Hash* (tabela de dispersão) é constituído de duas etapas principais:

- A chave de pesquisa é transformada em um endereço de uma tabela.
- Caso duas chaves sejam mapeadas na mesma posição, a **colisão**:
  - O dado pode ser rejeitado;
  - Uma estrutura auxiliar acomoda as duas chaves na mesma linha da tabela.

Hash table

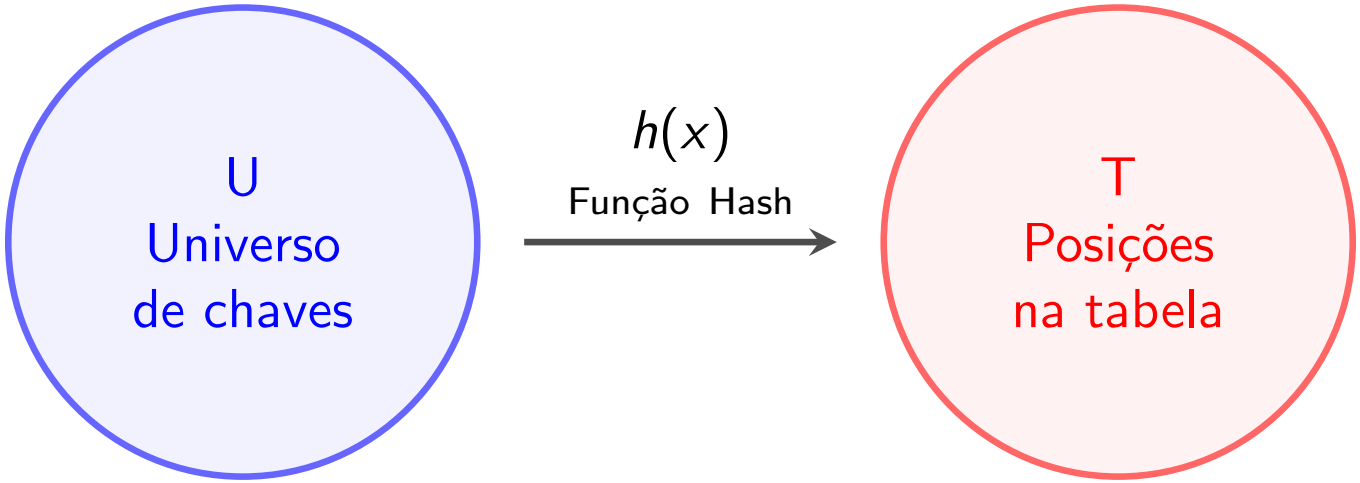
15	∅
14	∅
13	**61**
12	∅
11	∅
10	**90**
9	∅
8	∅
7	∅
6	∅
5	**21**
4	∅
3	∅
2	∅
1	∅
0	**32**

$$h(k) = mod(k, 16)$$

$$k \in \{61, 90, 21, 32\}$$

$$h(32) = mod(32, 16) = 0$$

Conceito de *Hash*



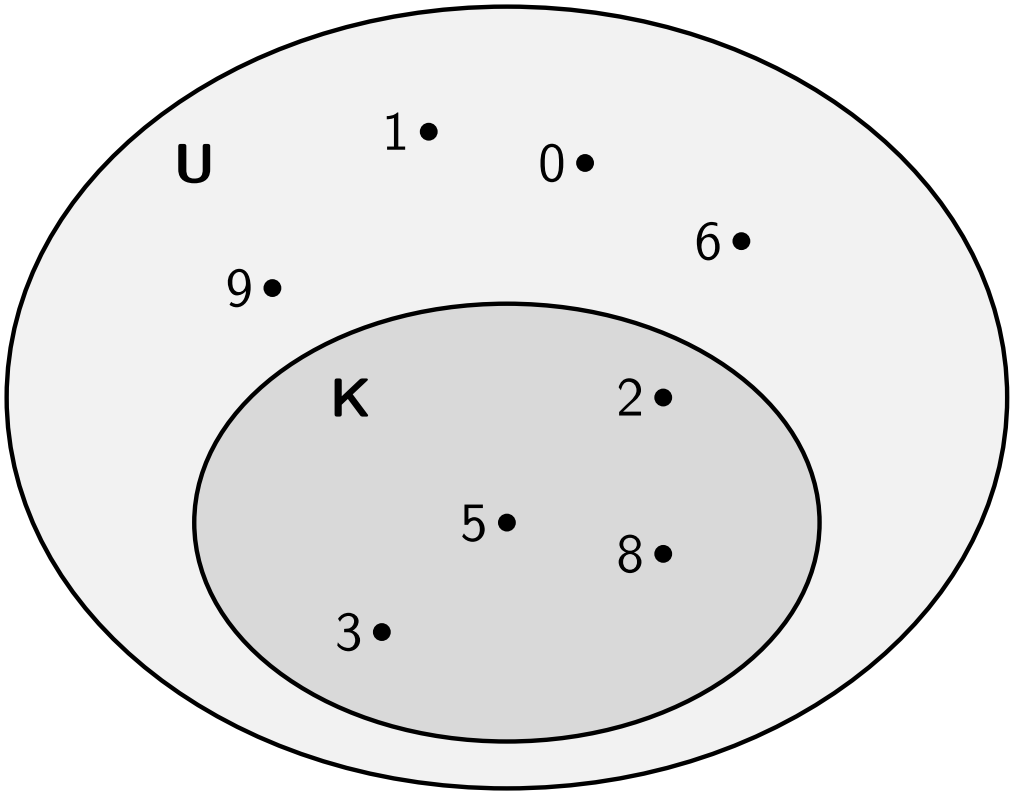
Endereçamento Direto

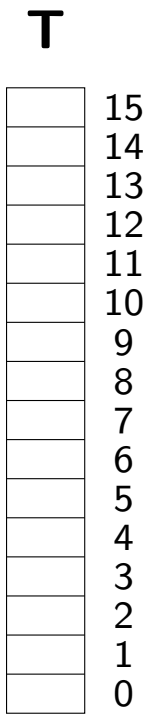
- Quando o universo de chaves **U** é pequeno, podemos alocar uma tabela com uma posição para cada chave. Ou seja:

$$|T| = |U|$$

- Logo, para cada posição da tabela, que pode ser implementada como um vetor, representa uma chave de **U** e armazena um elemento ou um ponteiro o elemento.

Endereçamento Direto





Endereçamento Direto

Situação prática:

[1] http://portal.mec.gov.br/ultimas-noticias/212-educacao-superior-1690610854/57981-ministro-inaugura-ampliacao-de-universidade-rural-do-rn

Endereçamento Direto

Situação prática:

- Contudo, geralmente $|U|$ **não** é pequeno.

[1] http://portal.mec.gov.br/ultimas-noticias/212-educacao-superior-1690610854/57981-ministro-inaugura-ampliacao-de-universidade-rural-do-rn

Endereçamento Direto

Situação prática:

- Contudo, geralmente $|U|$ **não** é pequeno.
- Exemplo: Matrícula do aluno da UFERSA.
  - 10 dígitos.
  - $|U| = 10^{10} = 10$ bilhões de combinações.

[1] http://portal.mec.gov.br/ultimas-noticias/212-educacao-superior-1690610854/57981-ministro-inaugura-ampliacao-de-universidade-rural-do-rn

Endereçamento Direto

Situação prática:

- Contudo, geralmente $|U|$ **não** é pequeno.
- Exemplo: Matrícula do aluno da UFERSA.
  - 10 dígitos.
  - $|U| = 10^{10} = 10$ bilhões de combinações.
- O primeiro dígito não pode ser 0 (zero), pois não seriam 10 dígtos.

[1] http://portal.mec.gov.br/ultimas-noticias/212-educacao-superior-1690610854/57981-ministro-inaugura-ampliacao-de-universidade-rural-do-rn

Endereçamento Direto

Situação prática:

- Contudo, geralmente $|U|$ **não** é pequeno.
- Exemplo: Matrícula do aluno da UFERSA.
  - 10 dígitos.
  - $|U| = 10^{10} = 10$ bilhões de combinações.
- O primeiro dígito não pode ser 0 (zero), pois não seriam 10 dígtos.
- Os quatro primeiros dígitos refere-se ao ano. Não existe muitas opções:

$$D_{14} = \{2005, 2006, \ldots 2021\}$$

[1] http://portal.mec.gov.br/ultimas-noticias/212-educacao-superior-1690610854/57981-ministro-inaugura-ampliacao-de-universidade-rural-do-rn

Endereçamento Direto

Situação prática:

- Contudo, geralmente $|U|$ **não** é pequeno.
- Exemplo: Matrícula do aluno da UFERSA.
  - 10 dígitos.
  - $|U| = 10^{10} = 10$ bilhões de combinações.
- O primeiro dígito não pode ser 0 (zero), pois não seriam 10 dígtos.
- Os quatro primeiros dígitos refere-se ao ano. Não existe muitas opções:

$$D_{14} = \{2005, 2006, \ldots 2021\}$$

- Mesmo assim, restam ainda: $16 * 10^6 = 16$ milhões de combinações!

[1] http://portal.mec.gov.br/ultimas-noticias/212-educacao-superior-1690610854/57981-ministro-inaugura-ampliacao-de-universidade-rural-do-rn

Endereçamento Direto

Situação prática:

- Contudo, geralmente $|U|$ **não** é pequeno.
- Exemplo: Matrícula do aluno da UFERSA.
  - 10 dígitos.
  - $|U| = 10^{10} = 10$ bilhões de combinações.
- O primeiro dígito não pode ser 0 (zero), pois não seriam 10 dígtos.
- Os quatro primeiros dígitos refere-se ao ano. Não existe muitas opções:

$$D_{14} = \{2005, 2006, \ldots 2021\}$$

- Mesmo assim, restam ainda: $16 * 10^6 = 16$ milhões de combinações!
- Em comparação, a A UFERSA tem 9.3 mil alunos presenciais em 2012[1].

[1] http://portal.mec.gov.br/ultimas-noticias/212-educacao-superior-1690610854/57981-ministro-inaugura-ampliacao-de-universidade-rural-do-rn

Dimensão da tabela *Hash*

- Deve ser escolhido adequadamente para diminuir o númer de colisões.

Dimensão da tabela *Hash*

- Deve ser escolhido adequadamente para diminuir o númer de colisões.
- Costuma ser um número primo.

Dimensão da tabela *Hash*

- Deve ser escolhido adequadamente para diminuir o númer de colisões.
- Costuma ser um número primo.
- Valores aceitáveis:

Dimensão da tabela *Hash*

- Deve ser escolhido adequadamente para diminuir o númer de colisões.
- Costuma ser um número primo.
- Valores aceitáveis:
  - A taxa de ocupação não deve ser maior que 75%.

Dimensão da tabela *Hash*

- Deve ser escolhido adequadamente para diminuir o númer de colisões.
- Costuma ser um número primo.
- Valores aceitáveis:
  - A taxa de ocupação não deve ser maior que 75%.
  - Uma taxa menor que 25% é considerado um gasto excessivo de memória.

Dimensão da tabela *Hash*

- Deve ser escolhido adequadamente para diminuir o númer de colisões.
- Costuma ser um número primo.
- Valores aceitáveis:
  - A taxa de ocupação não deve ser maior que 75%.
  - Uma taxa menor que 25% é considerado um gasto excessivo de memória.
  - Ideal: Próximo de 50%.

Exemplo de tabela *Hash*

Exemplo de tabela *Hash*

- Tipo **aluno** definido como estrutura e criado tipo **Hash** que é um vetor de ponteiros de alunos.

Exemplo de tabela *Hash*

- Tipo **aluno** definido como estrutura e criado tipo **Hash** que é um vetor de ponteiros de alunos.
- Função Hash indica onde guardar a os dados do aluno em função de sua matrícula.

```
1 #define N 255
2 typedef struct aluno {
3     int mat;
4     char nome[81];
5     char email[41];
6     char turma;
7 } * Hash[N];
```

Exemplo de tabela *Hash*

- Tipo **aluno** definido como estrutura e criado tipo **Hash** que é um vetor de ponteiros de alunos.
- Função Hash indica onde guardar a os dados do aluno em função de sua matrícula.

```c
#define N 255
typedef struct aluno {
    int mat;
    char nome[81];
    char email[41];
    char turma;
} * Hash[N];
```

```c
int hash(int mat){
    return mat % N;
}
```

Problemas

Problemas

- Qualquer valor maior que K é desperdício.

Problemas

- Qualquer valor maior que K é desperdício.
- Na prática, $|T| \gg |K|$ e os elementos de **K** não são conhecidos.

Problemas

- Qualquer valor maior que K é desperdício.
- Na prática, $|T| \gg |K|$ e os elementos de **K** não são conhecidos.
- Ao mapear os valores de **K** em **T**, mesmo com poucas chaves, pode ocorrer colisões.

Objetivos para Tabelas de Dispersão

Objetivos para Tabelas de Dispersão

- Desenvolver as funções de Hash;
- Compreender como ocorre o processo de colisão;
- Compreender o tratamento de colisões;
- Apresentar as melhores estuturas para construção de Tabelas de Dispersão.

Funções de dispersão

- Método da Divisão;
- Método da Dobra;
- Método da Multiplicação.

Método da divisão

Fácil, elegante e muito empregado;

Consiste em determina o resto da divisão entre a chave e o tamanho da tabela;

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

Método da divisão

Método da divisão

Função HASH:

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

Método da divisão

Função HASH:

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

- Alguns valores de **M** são maiores do que outro;

Método da divisão

Função HASH:

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

- Alguns valores de **M** são maiores do que outro;
- Se **M** é um número par, $h(x)$ será par juntamente com x. O mesmo ocorre com $h(x)$ ímpar. Não é um boa escolha.

Método da divisão

Função HASH:

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

- Alguns valores de **M** são maiores do que outro;
- Se **M** é um número par, $h(x)$ será par juntamente com x. O mesmo ocorre com $h(x)$ ímpar. Não é um boa escolha.
- Se **M** for uma potência de 2, $h(x)$ dependerá apenas de alguns dígitos de x. Pior situação.

Método da divisão

Função HASH:

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

- Alguns valores de **M** são maiores do que outro;
- Se **M** é um número par, $h(x)$ será par juntamente com x. O mesmo ocorre com $h(x)$ ímpar. Não é um boa escolha.
- Se **M** for uma potência de 2, $h(x)$ dependerá apenas de alguns dígitos de x. Pior situação.
- Alguns critérios para escolha:

Método da divisão

Função HASH:

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

- Alguns valores de **M** são maiores do que outro;
- Se **M** é um número par, $h(x)$ será par juntamente com x. O mesmo ocorre com $h(x)$ ímpar. Não é um boa escolha.
- Se **M** for uma potência de 2, $h(x)$ dependerá apenas de alguns dígitos de x. Pior situação.
- Alguns critérios para escolha:
  - Escolher **M** de modo que seja um número primo não próximo a uma potência de 2.
  - Escolher **M** tal que não possua divisores primos menores do que 20.

Método da dobra

- Suponha a chave como uma sequência de dígitos escritos num pedaço de papel.
- O método consiste em "dobrar"esse papel, de maneira que os dígitos se sobreponham.
- Estes devem ser somados sem considerar o "vai um".

Método da dobra

Exemplo: Chave 27 93 84

Método da dobra/Binário

Exemplo: Chave $71_{10} = 0001000111_2$

Método da Multiplicação

- Existem algumas variações.
- Variação mais conhecida: "Meio do quadrado":
  - A chave é multiplicada por ela mesma;
  - O resultado é armazenado em palavra de memória de **b** bits.
  - O número de escolhido para encontrar o endereço-base é retirado desse produto: Descartando os bits excessivos da extrema direita e da extrema esquerda.

Método da Multiplicação

Exemplo: Chaves do conjunto $[0, 127]$, mapeados para $[0, 15]$.