

Estrutura de Dados (CC4652)

Aula 9 - Tabela de Espalhamento (Hash)

Prof. Luciano Rossi

Ciência da Computação
Centro Universitário FEI

2º Semestre de 2023

Tabelas de Espalhamento (Hash)

Motivação: Estratégias de Busca

- Estrutura de lista linear sem ordenação (**Busca Linear**)

5	25	35	70	95	57	23	54	1
---	----	----	----	----	----	----	----	---

- Quanto tempo eu levaria para uma busca nesta lista, no pior caso?

Tabelas de Espalhamento (Hash)

Motivação: Estratégias de Busca

- Estrutura de lista linear com ordenação (**Busca Binária**):

1	5	23	25	35	54	57	70	95
---	---	----	----	----	----	----	----	----

- Quanto tempo eu levaria para uma busca nesta lista, no pior caso?
- Será que podemos fazer melhor?

Tabelas de Espalhamento (Hash)

Motivação: complexidade

- Complexidade de algoritmos de busca:

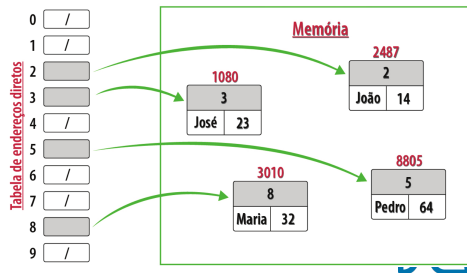
- ▶ Busca Linear: $O(n)$
- ▶ Busca Binária: $O(\log_2 n)$
- ▶ Hash: $O(1)$

Tabelas de Espalhamento (Hash)

Tabela de Endereçamento Direto

```
1 int main(void) {
2     registro *tabela = malloc(10 * sizeof(registro));
3     registro r1 = {2, "Joao", 14};
4     tabela[2] = r1;
5     registro r2 = {3, "Jose", 23};
6     tabela[3] = r2;
7     registro r3 = {5, "Pedro", 64};
8     tabela[5] = r3;
9     registro r4 = {8, "Maria", 32};
10    tabela[8] = r4;
11
12    for(int i = 0; i < 10; i++){
13        printf("ID: %d\n", i);
14        printf("Nome: %s\n", tabela[i].nome);
15        printf("Idade: %d\n", tabela[i].idade);
16        puts("");
17    }
18    return 0;
19 }
```

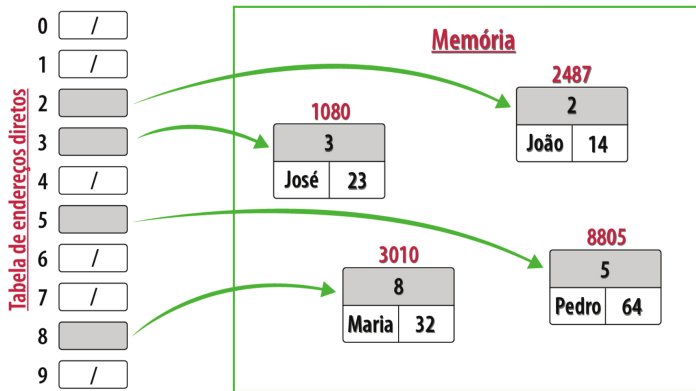
```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 typedef struct {
5     int idx;
6     char nome[15];
7     int idade;
8 } registro;
```



Tabelas de Espalhamento (Hash)

Tabela de Endereçamento Direto

1	ID: 0	ID: 5
2	Nome:	Nome: Pedro
3	Idade: 0	Idade: 64
4		
5	ID: 1	ID: 6
6	Nome:	Nome:
7	Idade: 0	Idade: 0
8		
9	ID: 2	ID: 7
10	Nome: Joao	Nome:
11	Idade: 14	Idade: 0
12		
13	ID: 3	ID: 8
14	Nome: Jose	Nome: Maria
15	Idade: 23	Idade: 32
16		
17	ID: 4	ID: 9
18	Nome:	Nome:
19	Idade: 0	Idade: 0



Tabelas de Espalhamento (Hash)

Tabela de Endereçamento Direto - Problemas

- Suponha que o conjunto de chaves a serem armazenadas seja $D = \{2, 100.002\}$.
- Nesse caso, se considerarmos o conjunto universo das chaves com limite superior igual ao maior valor de chave em D , teremos a implementação de um vetor com 100.002 posições, no qual apenas duas dessas posições serão associadas aos endereços de memória para as chaves 2 e 100.002.
- As demais 100.000 posições estarão associadas à *NULL*, isto é, estarão vazias.
- Nesse sentido, é fácil concluir que a utilização de uma Tabela de Endereço Direto não é uma solução viável quando temos um conjunto universo grande.

Tabelas de Espalhamento (Hash)

O que é uma Tabela Hash?

Definição

É uma estrutura de dados especial, que associa chaves de pesquisa a valores. Com a finalidade de promover uma busca eficiente.

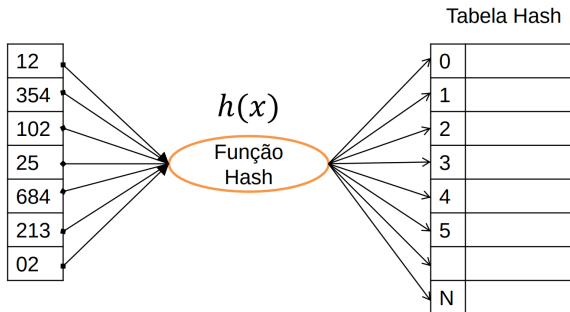
Tabelas de Espalhamento (Hash)

Ideia geral

- Cada chave ocupa a estrutura de dados através de uma função
- Esta função mapeia uma CHAVE para um valor INTEIRO
- Esta função se chama: função HASH.
- A estrutura de dados que abriga as chaves chama-se: Tabela Hash

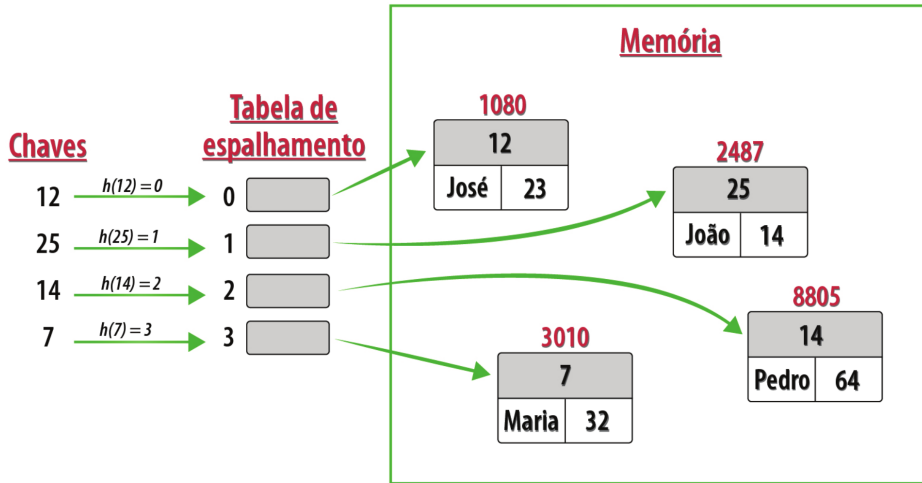
Tabelas de Espalhamento (Hash)

Ideia geral



Tabelas de Espalhamento (Hash)

Tabela Hash



Tabelas de Espalhamento (Hash)

Tabela Hash

- O processo considera um conjunto de chaves, com seus respectivos dados satélites, que devem ser armazenados em memória e os endereços devem ser mapeados em uma Tabela de Espalhamento (hash).
- Veja que os valores das chaves não correspondem aos índices das posições na tabela, assim, uma **função de espalhamento** cuidará para que cada valor de chave seja transformado em um índice existente na tabela.

Tabelas de Espalhamento (Hash)

Função de Espalhamento (Hash)

- Objetivo: Transformar uma chave (de qualquer tipo) em um número inteiro que representa o índice na tabela Hash
- Essa função deve ter as seguintes características:
 - ▶ Fácil de computar (rápida)
 - ▶ Criar uma distribuição equiprovável das chaves da tabela (depende do domínio e natureza da chave sendo computada)
 - ▶ Existem muitas funções Hash. Algumas são:
 - ★ Método da divisão
 - ★ Método da dobra
 - ★ Método da multiplicação
 - ★ Hashing Universal

Tabelas de Espalhamento (Hash)

Função Hash - Método da divisão - Exemplo

- Exemplos da função: $h(k) = k \bmod m$
- Suponha uma tabela hash de 10 posições ($m = 10$)
- Suponha que nossa chave (k) seja o próprio conteúdo a guardar

Item para guardar:

0	1	2	3	4	5	6	7	8	9

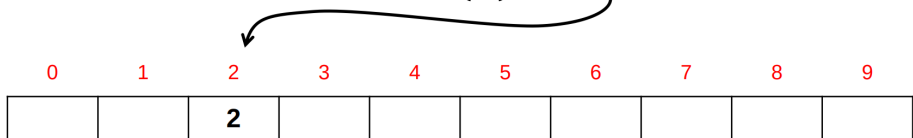
Tabelas de Espalhamento (Hash)

Função Hash - Método da divisão - Exemplo

- Exemplos da função: $h(k) = k \bmod m$
- Suponha uma tabela hash de 10 posições ($m = 10$)
- Suponha que nossa chave (k) seja o próprio conteúdo a guardar

Item para guardar: 2

$$h(2) = 2$$



0	1	2	3	4	5	6	7	8	9
		2							

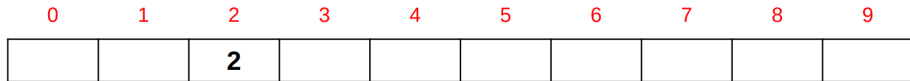
Tabelas de Espalhamento (Hash)

Função Hash - Método da divisão - Exemplo

- Exemplos da função: $h(k) = k \bmod m$
- Suponha uma tabela hash de 10 posições ($m = 10$)
- Suponha que nossa chave (k) seja o próprio conteúdo a guardar

Item para guardar: 7

$$h(7) = 7$$



0	1	2	3	4	5	6	7	8	9
		2							

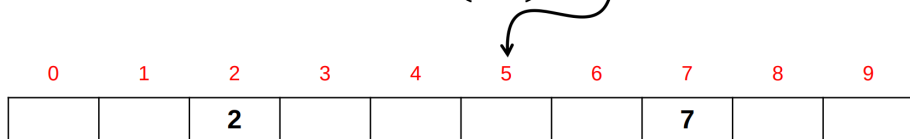
Tabelas de Espalhamento (Hash)

Função Hash - Método da divisão - Exemplo

- Exemplos da função: $h(k) = k \bmod m$
- Suponha uma tabela hash de 10 posições ($m = 10$)
- Suponha que nossa chave (k) seja o próprio conteúdo a guardar

Item para guardar: 15

$$h(15) = 5$$



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

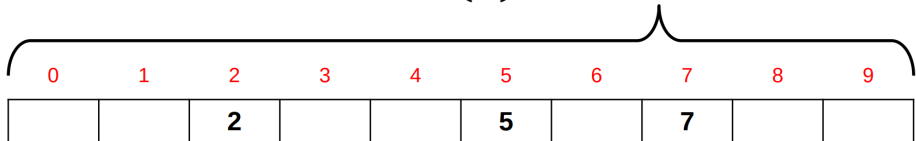
Tabelas de Espalhamento (Hash)

Função Hash - Método da divisão - Exemplo

- Exemplos da função: $h(k) = k \bmod m$
- Suponha uma tabela hash de 10 posições ($m = 10$)
- Suponha que nossa chave (k) seja o próprio conteúdo a guardar

Item para guardar: x

$$h(x) = x \bmod 10$$



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

Tabelas de Espalhamento (Hash)

Função Hash - Colisões

- Qualquer que seja a função de espalhamento, sempre há a possibilidade de colisões, que devem ser resolvidas, mesmo que se obtenha uma distribuição de registros de forma uniforme
- Tais colisões devem ser corrigidas de alguma forma
- O ideal seria uma função Hash, dada uma chave k , a probabilidade da função retornar $ik, 1 \leq ik \leq M$, fosse:

$$P(h(k) = ik) = \frac{1}{M}$$

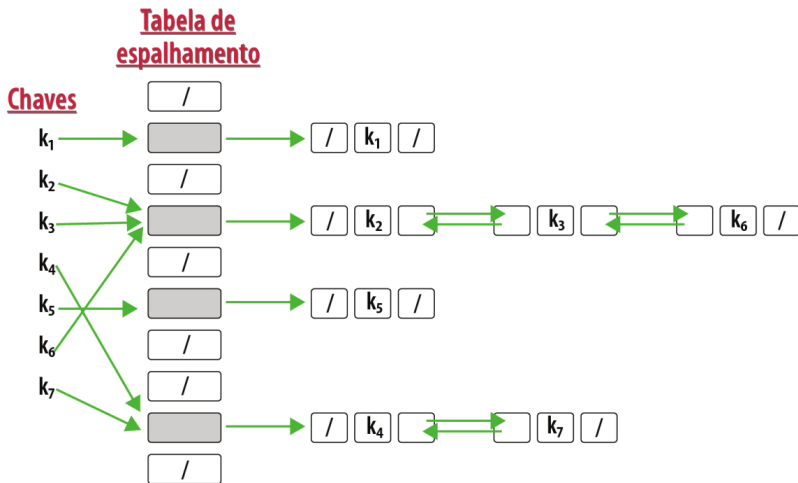
ou seja, sem colisões, tal função é difícil, se não impossível.

- Um agravante é que, para cada tipo de problema, existe uma função hash que melhor distribui as chaves.

Tabelas de Espalhamento (Hash)

Função Hash - Colisões - Encadeamento Externo ou Separado

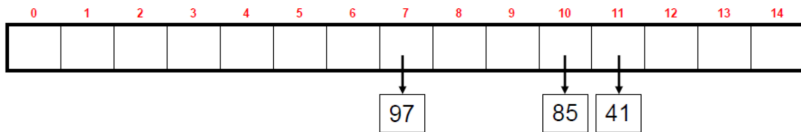
- Coloca-se chaves com $h(k)$ iguais em uma lista encadeada.



Tabelas de Espalhamento (Hash)

Função Hash - Colisões - Encadeamento Externo ou Separado - Exemplo

Exemplo de tratamento de colisão empregando o método do encadeamento externo e o da divisão:



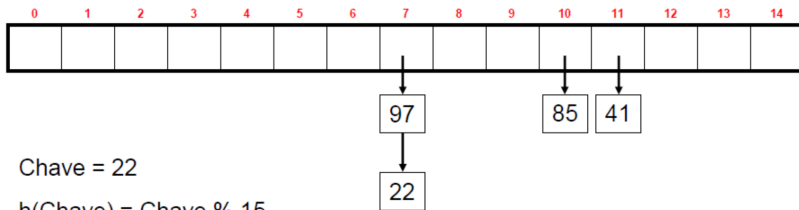
Chave = __

$h(\text{Chave}) = \text{Chave} \% 15$

Tabelas de Espalhamento (Hash)

Função Hash - Colisões - Encadeamento Externo ou Separado - Exemplo

Exemplo de tratamento de colisão empregando o método do encadeamento externo e o da divisão:



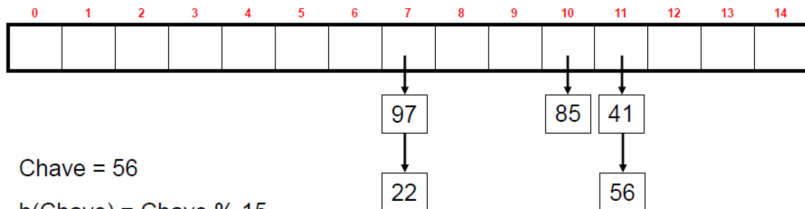
Chave = 22

$$\begin{aligned} h(\text{Chave}) &= \text{Chave} \% 15 \\ &= 22 \% 15 = \mathbf{7} \end{aligned}$$

Tabelas de Espalhamento (Hash)

Função Hash - Colisões - Encadeamento Externo ou Separado - Exemplo

Exemplo de tratamento de colisão empregando o método do encadeamento externo e o da divisão:



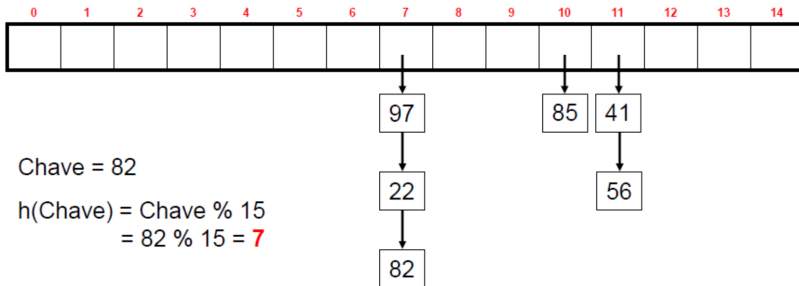
Chave = 56

$$\begin{aligned} h(\text{Chave}) &= \text{Chave} \% 15 \\ &= 56 \% 15 = \mathbf{11} \end{aligned}$$

Tabelas de Espalhamento (Hash)

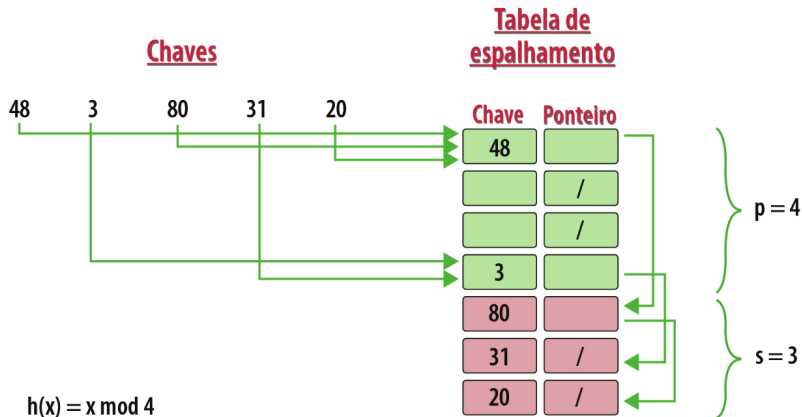
Função Hash - Colisões - Encadeamento Externo ou Separado - Exemplo

Exemplo de tratamento de colisão empregando o método do encadeamento externo e o da divisão:



Tabelas de Espalhamento (Hash)

Função Hash - Colisões - Tratamento de colisão por encadeamento interior



Tabelas de Espalhamento (Hash)

Função Hash - Colisões - Tratamento de colisão por Endereçamento Aberto

- Também conhecido como Rehash ou Reespalhamento.
- Trata-se de uma função de reespalhamento $rh()$, que aceita um índice de vetor e gera outro.
- Ou seja, se $h(k)$ está ocupado, $rh()$ é aplicado sobre $h(k)$, $rh(h(k))$.
- Que por sua vez, se estiver ocupado, $rh()$ é aplicado sobre $rh(h(k))$, ou seja, $rh(rh(h(k)))$ e assim opera até encontrar uma posição livre.

Tabelas de Espalhamento (Hash)

Função Hash - Colisões - Tratamento de colisão por Endereçamento Aberto

- Endereçamento Aberto: Espalhamento Linear
- Também conhecida como Overflow Progressivo
- Define a função de reespalhamento como a busca da próxima posição livre, ou seja: $rh_1(k) = (h(k) + 1) \bmod M$
- No caso da próxima posição estar ocupada, busca-se a próxima da próxima e assim por diante.

$$rh_2(k) = rh_1(h(k) + 1) \bmod M$$

$$rh_2(k) = (h(k) + 1 + 1) \bmod M$$

$$rh_2(k) = (h(k) + 2) \bmod M$$

$$rh_3(k) = rh_2(rh_1(h(k) + 1)) \bmod M$$

$$rh_3(k) = h(k) + 2 + 1 \bmod M$$

$$rh_3(k) = h(k) + 3 \bmod M$$

...

$$rh_n(k) = h(k) + n \bmod M$$

Tabelas de Espalhamento (Hash)

Função Hash - Colisões - Tratamento de colisão por Endereçamento Aberto - Exemplo

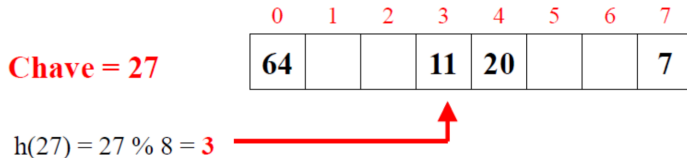
- Endereçamento Aberto: Espalhamento Linear

Chave =

0	1	2	3	4	5	6	7
64			11	20			7

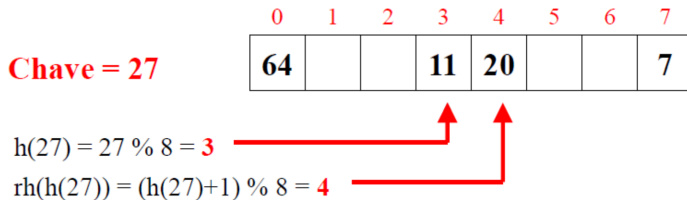
Tabelas de Espalhamento (Hash)

Função Hash - Colisões - Tratamento de colisão por Endereçamento Aberto - Exemplo



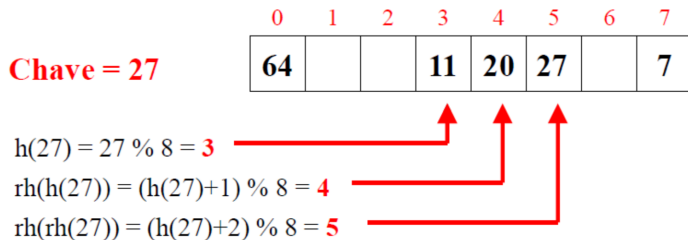
Tabelas de Espalhamento (Hash)

Função Hash - Colisões - Tratamento de colisão por Endereçamento Aberto - Exemplo



Tabelas de Espalhamento (Hash)

Função Hash - Colisões - Tratamento de colisão por Endereçamento Aberto - Exemplo



Tabelas de Espalhamento (Hash)

Exemplo

- Função hash: método da divisão;
- Tratamento de colisões: encadeamento externo.
- Conjunto de valores: {190, 322, 172, 89, 13, 4, 769, 61, 15, 76, 97, 28, 80, 76, 88}
- Passos:
 1. A tabela hash será um arranjo, de tamanho igual à quantidade de valores esperada, no qual cada posição é um LDE;
 2. Inicialize cada posição da hash com uma LDE vazia;
 3. Construa um procedimento para inserir um valor na hash;
 4. Construa um procedimento para remover um valor da hash.

Estrutura de Dados (CC4652)

Aula 9 - Tabela de Espalhamento (Hash)

Prof. Luciano Rossi

Ciência da Computação
Centro Universitário FEI

2º Semestre de 2023