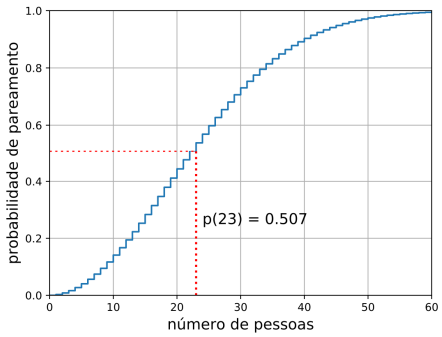

# Tabelas Hash

Prof. Marcos Carrard
carrard@univali.br
carrard@gmail.com

UNIVALI Escola Politécnica

# Paradoxo do Aniversário




$$\frac{364}{365} \cdot \frac{363}{365} \cdot \ldots \cdot \frac{365 - n + 1}{365} < \frac{1}{2}$$

Richard von Mises (1939(

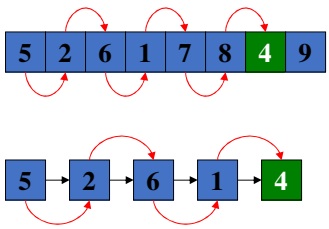
https://super.abril.com.br/ideias/newsletter-o-paradoxo-do-aniversario-puzzle-de-08-02-2018/

UNIVALI Escola Politécnica

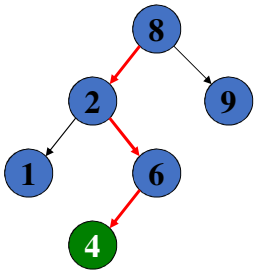
# Motivação

- Dada uma tabela com uma chave e vários valores por linha, quero rapidamente procurar, inserir e apagar registros baseados nas suas chaves
- Estruturas de busca sequencial/binária levam tempo até encontrar o elemento desejado.

Ex: Arrays e listas




Ex: Árvores




UNIVALI Escola Politécnica

# Motivação

Suponha que você pudesse criar um array onde qualquer item pudesse ser localizado através de acesso direto.

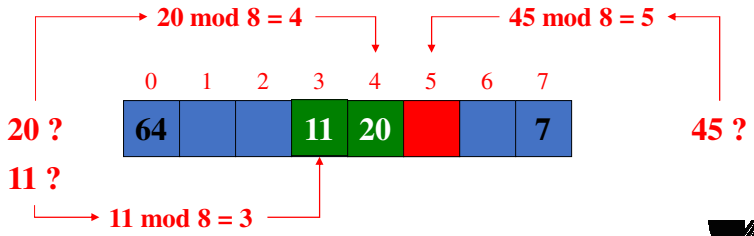
Isso seria ideal em aplicações do *tipo Dicionário*, onde gostaríamos de fazer consultas aos elementos da tabela em tempo constante.

Ex: Tabela de símbolos em compiladores.

UNIVALI Escola Politécnica

# Motivação

- Em algumas aplicações, é necessário obter o valor com poucas comparações, logo, é preciso saber a posição em que o elemento se encontra, sem precisar varrer todas as chaves.
- A estrutura com tal propriedade é chamada de **tabela *hash***.




UNIVALI Escola Politécnica

# O Tamanho de uma tabela

Um problema é que – como o *espaço de chaves*, ou seja, o número de *possíveis* chaves, é muito grande – este array teria que ter um tamanho muito grande.

Ex: Se fosse uma tabela de nomes com 32 caracteres por nome, teríamos $26^{32} = (2^5)^{32} = 2^{160}$ possíveis elementos.

Haveria também o desperdício de espaço, pois a cada execução somente uma pequena fração das chaves estarão de fato presentes.

UNIVALI Escola Politécnica

# Para que serve *Hashing*?

O objetivo de *hashing* é mapear um espaço enorme de chaves em um espaço de inteiros relativamente pequeno.

Isso é feito através de uma função chamada *hash function*.

O inteiro gerado pela *hash function* é chamado *hash code* e é usado para encontrar a localização do item.

UNIVALI Escola Politécnica

# Funções *Hashing*

- Método pelo qual:
  - As chaves de pesquisa são transformadas em endereços para a tabela (função de transformação);
  - Obtém-se valor do endereço da chave na tabela HASH
- Tal função deve ser fácil de se computar e fazer uma distribuição equiprovável das chaves na tabela
- A essa função dá-se o nome de Função *HASHING*

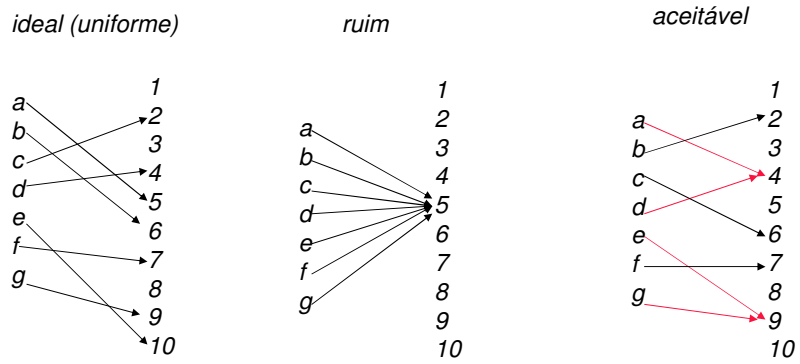
UNIVALI Escola Politécnica

# Funções *Hashing*

- Seja *M* o tamanho da tabela:
  - A função de hashing mapeia as chaves de entrada em inteiros dentro do intervalo [1..*M*]
- Formalmente:
  - A função de hashing $h(k_j) \rightarrow [1,M]$ recebe uma chave $k_j \in \{k_0,..,k_m\}$ e retorna um número *i*, que é o índice do subconjunto $m_i \in [1,M]$ onde o elemento que possui essa chave vai ser manipulado

UNIVALI Escola Politécnica

# Funções *Hashing*




UNIVALI Escola Politécnica

# Funções *Hashing*

- Existem várias funções *Hashing*, dentre as quais:
  - Resto da Divisão
  - Meio do Quadrado
  - Método da Dobra
  - Método da Multiplicação

UNIVALI Escola Politécnica

# Resto da Divisão

- Forma mais simples e mais utilizada
  - Nesse tipo de função, a chave é interpretada como um valor numérico que é dividido por um valor
- O endereço de um elemento na tabela é dado simplesmente pelo resto da divisão da sua chave por *M* ($F_h(x) = x \bmod M$), onde M é o tamanho da tabela e x é um inteiro correspondendo à chave

  0 <= F(x) <= M

UNIVALI Escola Politécnica

# Resto da Divisão – Desvantagens

- Função extremamente dependente do valor de M escolhido
  - M deve ser um número primo
  - Valores recomendáveis de M devem ser >20

UNIVALI Escola Politécnica

# Colisões

- Seja qual for a função, na prática existem **sinônimos** – chaves distintas que resultam em um mesmo valor de *hashing*.
- Quando duas ou mais chaves sinônimas são mapeadas para a mesma posição da tabela, diz-se que ocorre uma **colisão**.

UNIVALI Escola Politécnica

# Colisões

- Qualquer que seja a função de transformação, existe 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 tal que, dada uma chave 1 <= I <= 26, a probabilidade da função me retornar a chave x seja PROB($F_h$(x)= I) = 1/26, ou seja, não tenha colisões, mas tal função é difícil, se não impossível

UNIVALI Escola Politécnica

# Tratamento de Colisões

- Alguns dos algoritmos de Tratamento de Colisões são:
  - Endereçamento Fechado ou Externo
  - Endereçamento Aberto ou Interno

UNIVALI Escola Politécnica

# Endereçamento Fechado

- Também chamado de *Overflow* Progressivo Encadeado
- Algoritmo: usar uma lista encadeada para cada endereço da tabela
- Vantagem: só sinônimos são acessados em uma busca. Processo simples.
  - Desvantagens:
    - É necessário um campo extra para os ponteiros de ligação.
    - Tratamento especial das chaves: as que estão com endereço base e as que estão encadeadas

UNIVALI Escola Politécnica

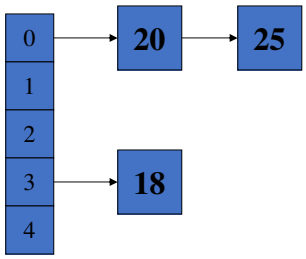
# Endereçamento Fechado

No **endereçamento fechado**, a posição de inserção não muda. Todos devem ser inseridos na mesma posição, através de uma **lista ligada** em cada uma.

**20 mod 5 = 0**

**18 mod 5 = 3**

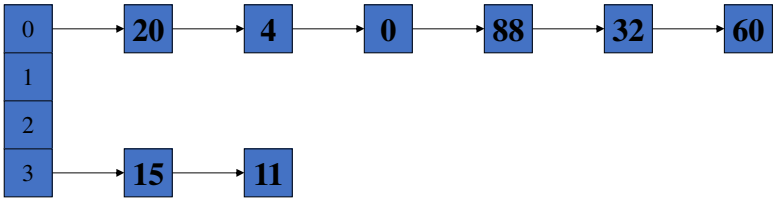
**25 mod 5 = 0**
colisão com 20




UNIVALI Escola Politécnica

# Endereçamento Fechado

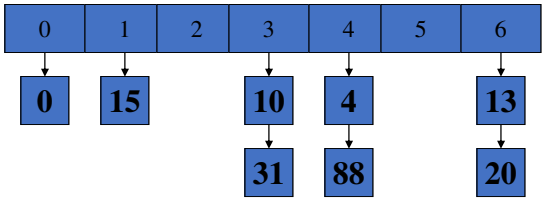
A busca é feita do mesmo modo: calcula-se o valor da função *hash* para a chave, e a busca é feita na lista correspondente.
Se o tamanho das listas variar muito, a busca pode se tornar ineficiente, pois a busca nas listas se torna sequencial




UNIVALI Escola Politécnica

# Endereçamento Fechado

É obrigação da função HASH distribuir as chaves entre as posições de maneira **uniforme**




UNIVALI Escola Politécnica

# Endereçamento Fechado

- Trata-se da colisão dentro da própria tabela
- Uma solução seria procurar a próxima posição vaga ou,
- Mudar sucessivamente a função de calculo, de forma controlada, para ir alocando em outro lugar da tabela.
- Podemos também criar, dentro da mesma tabela, uma zona de colisões, onde serão alocados sequencialmente ou não, as chaves colididas.

UNIVALI Escola Politécnica


K-HARD

UNIVALI Escola Politécnica