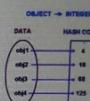
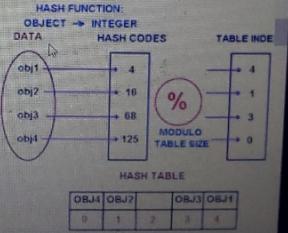


Hashing - Exemplo

Consideremos uma tabela hash com as seguintes características:



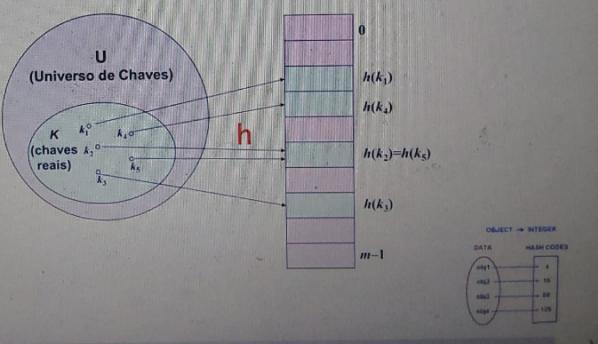
- A tabela terá um máximo de **80** entradas (**0..79**).
- Cada entrada da tabela irá armazenar o nome de um estudante.
- Os valores dos campos chave **k** terão valores contidos no intervalo [**0..1000**]
- A função **HASH** é definida por $h(k) = k \% 80$.
- Ou seja, a função $h(k)$ recebe uma chave **k** (entre 0 e 1000) e retorna um valor (índice) entre 0 e 79.



Topics Avançados de Estrutura de Dados - Unidade 15 - Hashing

Simulações da Função Hash

Key	383	487	235	527	510	564	103	66	14
Índice	63	7	75	47	30	4	23	66	14

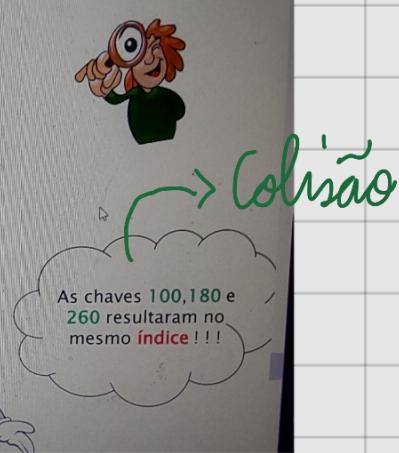


Hashing - Observação

No exemplo anterior, vamos considerar as seguintes chaves:

Key	100	180	260
Índice	20	20	20

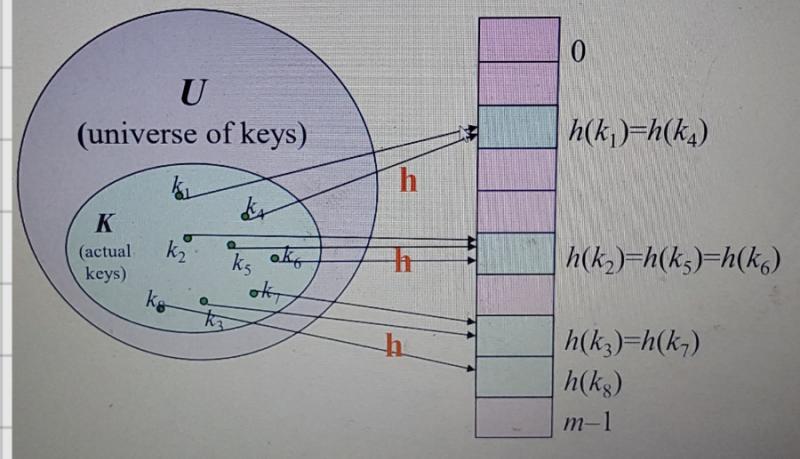
Função HASH: $h(k) = k \% 80$.



Tratamento da Colisão na Tabela Hash

- Encadeamento
- Endereço Aberto (Rehashing)

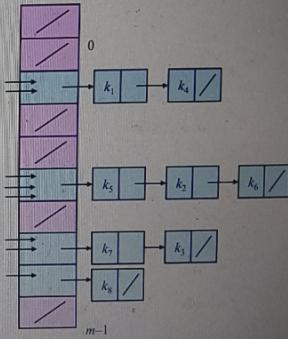
Resolução da Colisão por Encadeamento



Hashing com Encadeamento

- ⊕ Todos os elementos que têm mesmo hashcode são armazenados num slot que referencia uma lista encadeada (ligada).

- ⊕ O pointer para o head da lista é armazenado no slot da tabela hash.



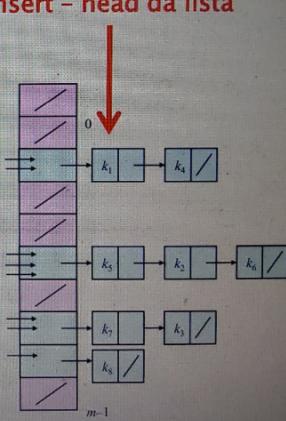
Hashing com Encadeamento Operações de Dicionário

Insert - head da lista

- ⊕ Operação de Insert no head da lista:

$T[h(\text{key}[x])]$

- ⊕ Complexidade no pior caso => $O(1)$



Hashing com Encadeamento Operações de Dicionário

Delete numa posição x

- ⊕ Operação de Delete numa posição qualquer da lista encadeada:

$T[h(\text{key}[x])]$

- ⊕ Complexidade no pior caso => Proporcional ao tamanho da lista: $O(n)$

