alpha ed/tech

ESTRUTURA DE DADOS

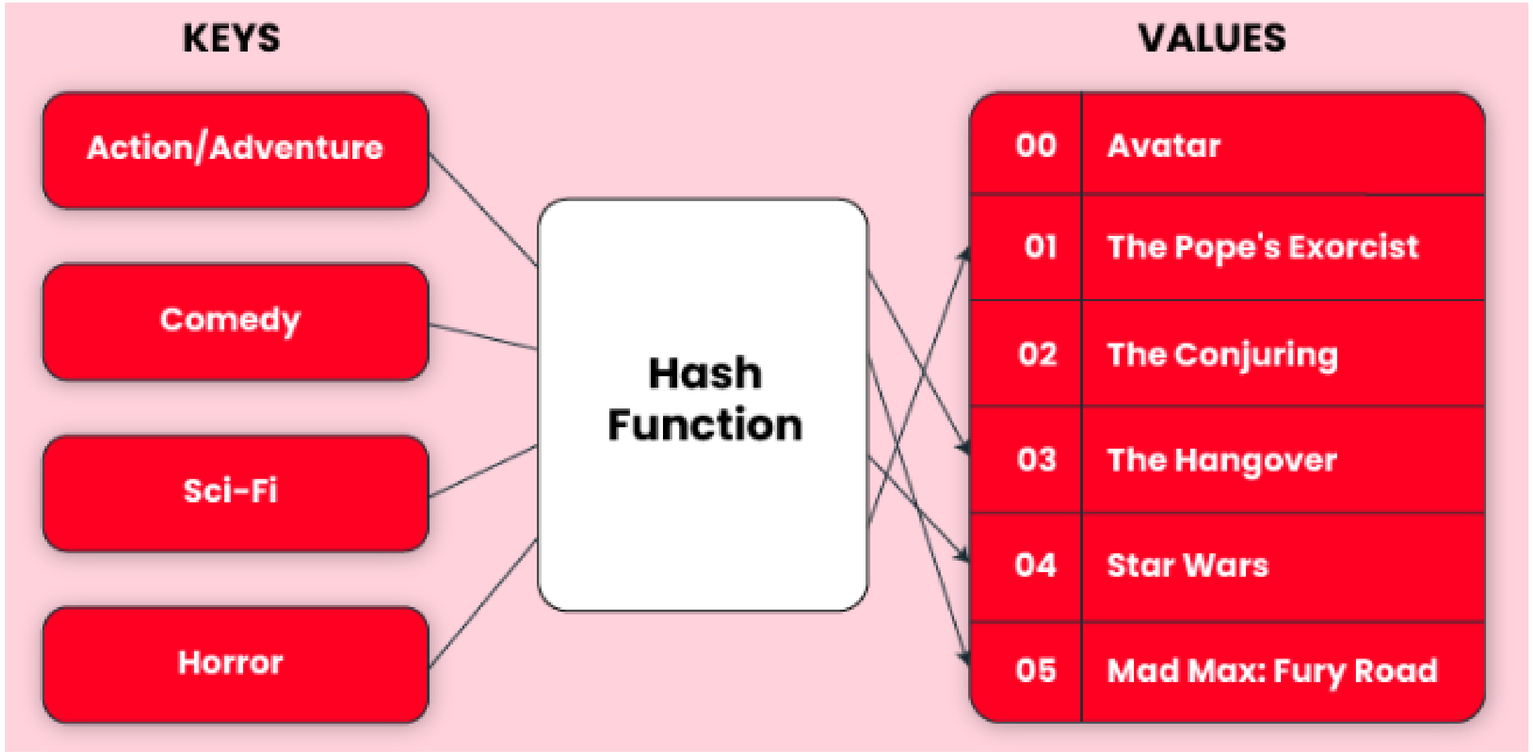
MÓDULO 07 | AULA 11

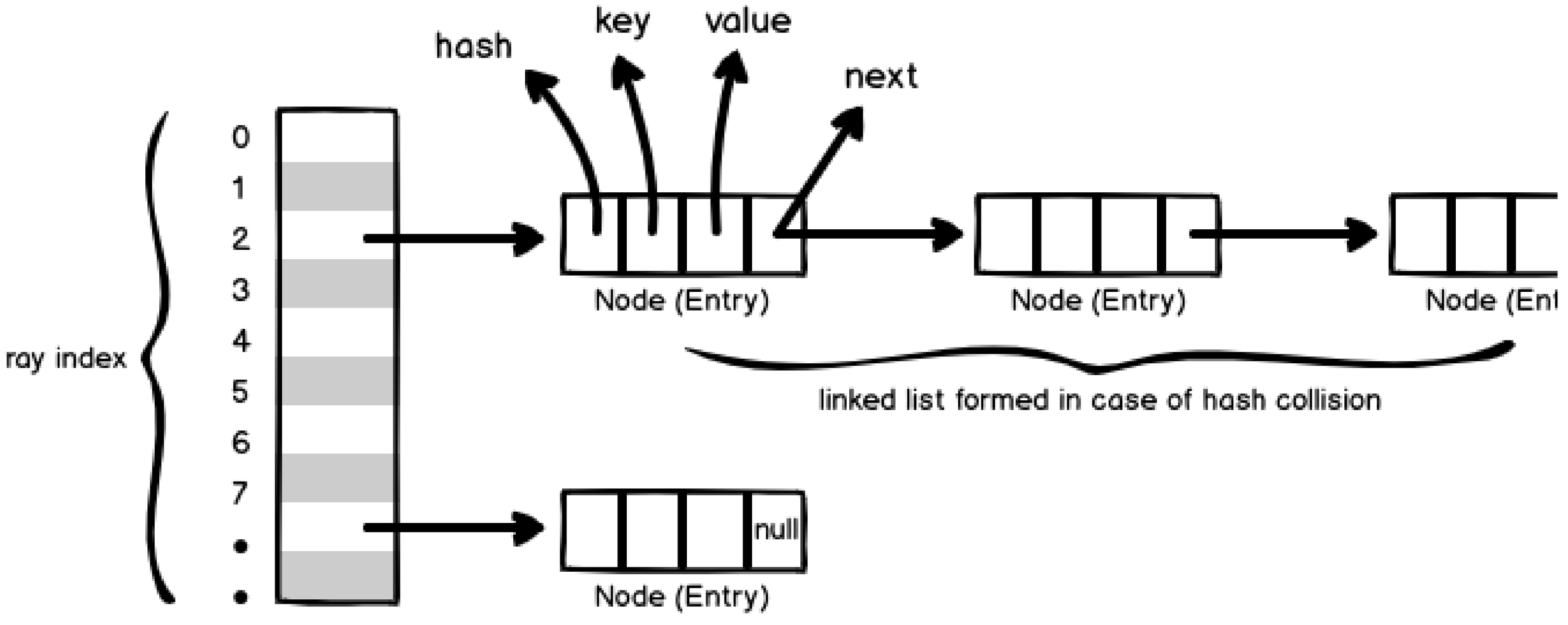
alpha
<ed/tech>

HASHMAP

Introdução

Hashmaps, também conhecidos como dicionários em algumas linguagens de programação, são estruturas de dados que armazenam pares de chave-valor, permitindo a rápida recuperação de um valor associado a uma chave específica. O sucesso desta estrutura reside na **função hash**, que converte a chave em um índice numérico, apontando para onde o valor está armazenado no array subjacente.




hash
key
value
next
0
1
2
3
4
5
6
7
ray index
Node (Entry)
Node (Entry)
Node (Ent
linked list formed in case of hash collision
null
Node (Entry)
Bucket (array) / Entry table

HashMap

alpha
<ed/tech>

HASHMAP

Como Funciona

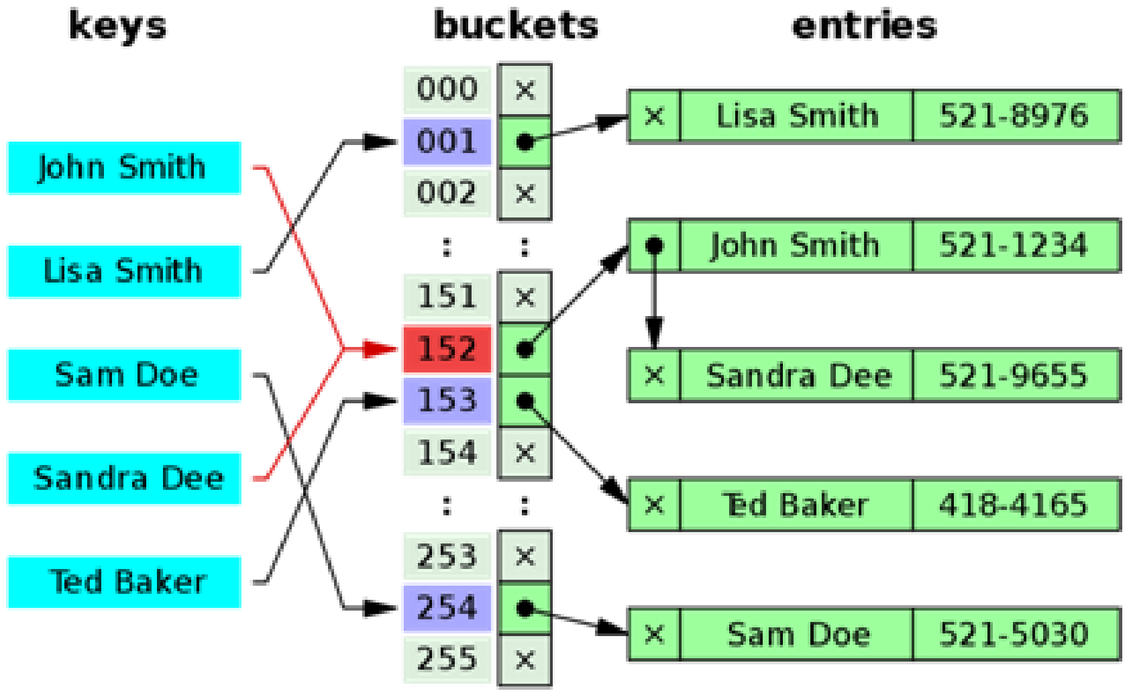
1. **Função Hash:** Recebe uma chave como entrada e retorna um índice no array. Esta função precisa ser rápida e distribuir uniformemente as chaves para minimizar colisões.
2. **Armazenamento de Valor:** O valor associado à chave é armazenado no índice determinado pela função hash.
3. **Recuperação de Valor:** Para recuperar um valor, a função hash calcula o índice usando a chave e acessa diretamente o array para encontrar o valor.

Pense em um hashmap como uma biblioteca e a função de hash como um bibliotecário. Quando você quer encontrar um livro (valor) específico, você dá o título do livro (chave) ao bibliotecário. O bibliotecário usa um sistema especial (função de hash) para encontrar rapidamente exatamente onde o livro está na biblioteca, sem ter que procurar em todas as prateleiras. Ou seja, nesse sentindo uma função de hash é uma função que pega uma entrada (uma chave) e retorna um valor de hash, que é um número. Este valor de hash é usado para determinar a posição exata (por exemplo, um índice) no array interno onde o valor associado à chave será armazenado.

alpha <ed/tech>

HASHMAP

A estrutura de um Hashmap geralmente é simples. Temos uma entrada (geralmente uma chave), a função hash calcula um valor hash para a chave que foi dada como entrada que leva a um índice de um array. Cada índice do array mencionado anteriormente leva a um “balde (bucket)”, que é essencialmente um “slot” ou “compartimento” dentro de um hashmap onde os elementos são armazenados, ou seja, Um balde é onde os pares chave-valor efetivamente são armazenados. Em termos simples, um hashmap é um array de baldes.

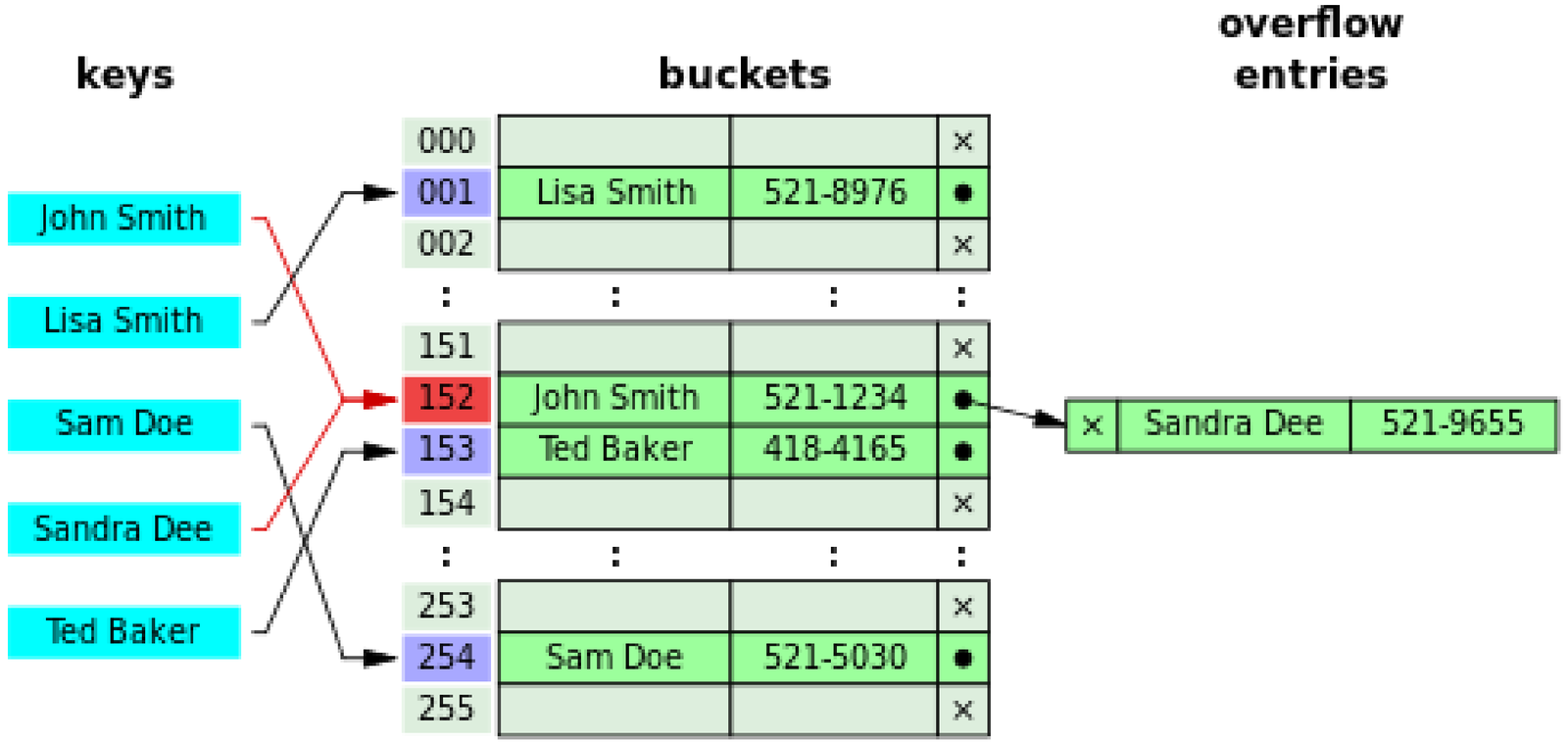


alpha
<ed/tech>

HASHMAP

Colisões em Hashmaps

Uma colisão ocorre quando duas chaves distintas são mapeadas para o mesmo índice pelo processo de hash. Colisões são inevitáveis em qualquer sistema de hashmap devido ao princípio da gaveta (ou Princípio da Casa dos Pombos), onde se tem mais chaves do que espaços de armazenamento.



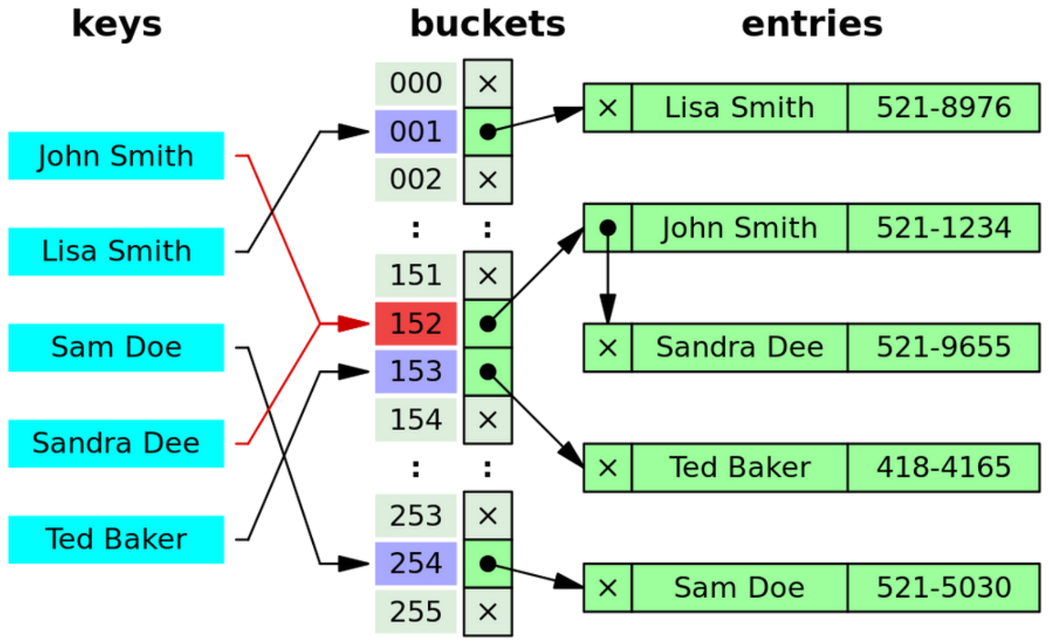
alpha <ed/tech>

HASHMAP

SOLUÇÕES PARA COLISÕES

Encadeamento

- Descrição: Cada índice do array armazena uma lista encadeada (ou outra estrutura de dados) de todos os elementos que são mapeados para aquele índice.
- Implementação: Se uma colisão ocorre, o novo valor é simplesmente adicionado à lista no respectivo índice.
- Vantagem: Simples de implementar.
- Desvantagem: O tempo de busca pode aumentar se muitos elementos forem mapeados para o mesmo índice.



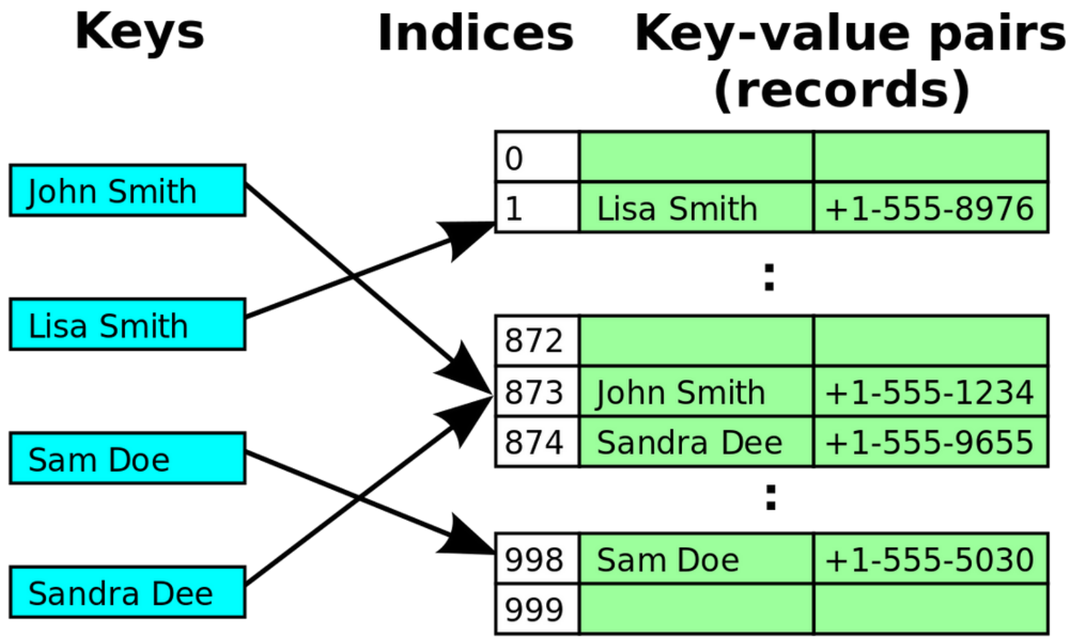
alpha <ed/tech>

HASHMAP

SOLUÇÕES PARA COLISÕES

Endereçamento Aberto

- Descrição: Todas as entradas são armazenadas no próprio array. Se ocorrer uma colisão, a estrutura procura o próximo índice vazio seguindo uma sequência predeterminada.
- Técnicas Comuns: Sondagem linear, sondagem quadrática, e hash duplo.
- Vantagem: Evita o uso de estruturas de dados adicionais.
- Desvantagem: O array pode ficar saturado, e a sondagem linear pode levar a aglomeração.



alpha <ed/tech>

HASHMAP

SOLUÇÕES PARA COLISÕES

Rehashing

- Descrição: Aumenta o tamanho do array de armazenamento e recalcula o índice de todos os elementos usando uma nova função hash.
- Implementação: É geralmente acionado quando o fator de carga (número de elementos armazenados / número de slots disponíveis) atinge um certo limite.
- Vantagem: Reduz a probabilidade de colisões e pode melhorar o desempenho de busca.
- Desvantagem: Pode ser uma operação custosa, pois requer recalculação e realocação de todos os elementos.