

Tabela Hash

Em uma faculdade é criado um **Banco de Dados** com o **RA** e o **Nome** de todos os alunos. Como é possível criar uma busca eficiente neste **Banco de Dados**?

Tipos de busca → sequencial → Percorre toda
→ binária → Precisamos de dados
ordenados
↳ ordenar é custoso.

Como funciona a hash

	Chave	Valor	Índice
qualquer tipo Template < T		0	
		1	
		2	
		3	

função hash

↳ Criar um {chave}{valor}
↳ busca a partir de
uma função hash.

função hash / espalhamento / Mapeamento

↳ pegar o Valor da chave e
transformar em um inteiro para
decidir qual Valor / endereço
a chave terá.

o problema a ser resolvido não
as colisões. Quando a função
hash retorna o mesmo Valor para
inputs diferentes.

FUNÇÃO HASH FUNÇÃO DE ESPALHAMENTO		CHAVE	VALOR
Quantidade de Letras - 1			0
MARIA			1
			2
			3
COLISÃO		PAULA	4

Regras para uma boa func hash

1 → Poucas Colisões

2 → Facilmente computável / processável

Não existe tabela hash sem colisões

↳ então como tratar colisões

Encadeamento Exterior / separado

↳ uso de listas encadeadas

Encadeamento Interior / aberto

↳ Heterogêneo.

↳ Homogêneo.

Encadeamento exterior

Chave

(set, float, int)

func hash

Valor int.

Exemplo de func

hash para int

Chave % tamanho

, resto

$$21 \% 6 = 3$$

$$9 \% 6 = 3$$

Valor

0

1

2

3

4

5

Chave

□

17

6

□

□

□

□

□

} Ponteiro para lista encadeada.

21

9

linked list

Como fica as três operações

inserção → func + adicionar lista

busca → passar pela func hash e percorrer a list

remoção → buscar e remover da lista encadeada

Encadeamento interior heteragêneo.

Valor	Chara	resto	função	Valor % 6
0	12			
1				
2			para buscar em	
3	21		caso de colisão	
4	20		é feita uma comparação	
5			com todo o espaço de colisão	
6	9			
7	33			
8				
9				

} heteragêneo porque separa
charas e colisão.

obs: para manter mesmo tipo, precisa diferenciar Vazio de
removido, pois se não o algoritmo vai parar a busca
antes da hora.

Homogeneo.

Valor	Chara	resto	função	Valor % 6
0				
1	7			
2	13			
3	20			
4				
5	11			

} linear para
estatístico
Muito

Colisão diferente
faz numerosa ocuparem
posição "errada"

obs: para manter mesmo tipo, precisa diferenciar Vazio de
removido, pois se não o algoritmo vai parar a busca
antes da hora.

obs - 2: Mesmo tipo o fator de carga elementos ímpar, pois
um fator alto/poucas posições vazias causa ↑ vetor
lentidão

além disso, um fator muito baixo significa memória sendo desperdiçada.