



Tabelas de Espalhamento (Hashing)

João Pedro de F. Lourenço
Leandro R. de Souza
Thiago B. Carnaíba
Vinícius A. Marins

joao.franca@aluno.ifsp.edu.br
r.leandro@aluno.ifsp.edu.br
thiago.bruchmann@aluno.ifsp.edu.br
v.marins@aluno.ifsp.edu.br

Discentes do Curso Bacharelado a Ciência da
Computação – IFSP – Campus Presidente
Epitácio

> Introdução

- Uma tabela de dispersão, chamada de tabela hash, é uma forma de armazenar dados para que possam ser rapidamente localizado, sendo assim, um vetor onde cada uma das posições armazena zero, uma, ou mais chaves (e valores associados).
- A idéia central do Hash é utilizar uma função, aplicada sobre parte da informação (chave), para retornar o índice onde a informação deve ou deveria estar armazenada.
- Melhor Caso: $O(1)$
- Caso médio: $O(1)$
- Pior caso: $O(n)$



> Motivação

- Métodos de busca vistos até agora utilizam vetores ordenados para realizar as operações.
- Ordenar um vetor e mantê-lo ordenado durante suas operações (inserção, edição e remoção) é computacionalmente custoso.
- Caso deseje buscar o índice a qual pertence um valor é necessário percorrer o vetor procurando a posição.
- O ideal é utilizar a informação para adquirir o índice do vetor.
- Possuir o tempo médio de $O(1)$.



> Função Hash

- Transformar a chave de pesquisa em um endereço da tabela.
- Parâmetros importantes:
 - M : número de posições na tabela de hash
 - N : número de chaves da tabela de símbolos
 - $\alpha = N/M$: fator de carga (load factor)
- Índice de inserção = $K \bmod M$
 - K é a chave de pesquisa.
 - M é o número de posições na tabela de hash
- Chaves diferentes podem gerar o mesmo índice (Colisão).



> Exemplo

0	1	2	3	4	5	6	7	8	9
			3		15				

Chave	Índice
15	5
23	3

Recomendação: M deve ser um número primo, devem ser evitados os números primos obtidos a partir de $b \cdot i \pm j$ onde b é a base do conjunto de caracteres (128 para ASCII por exemplo), e i e j são pequenos inteiros.

> Colisão

- O Paradoxo do Aniversário (Fellas, 1968)
 - Em um grupo de 23 ou mais pessoas, existe uma chance maior do que 50% de que 2 pessoas comemorem o aniversário no mesmo dia.
- Uma colisão ocorre quando a função hash gera o mesmo valor (índice) para duas chaves diferentes.
- Um hash perfeito seria uma função Bijetora.
- A maioria das funções são Sobrejetoras.
- A probabilidade (P) de se inserir N itens consecutivos sem colisão em uma tabela de tamanho M é $\frac{M!}{(M-N)!M^N}$

N	P
10	0,883
22	0,524
23	0,493
30	0,303

> Tratando Colisões

- Encadeamento
 - Cada entrada na tabela aponta para uma lista encadeada
 - Ao ocorrer uma colisão, é gerada uma nova entrada na respectiva lista.
 - O tempo de busca será constante somente se o número de elementos em cada lista for pequeno.
 - Caso a tabela fique muito desbalanceada, é interessante realizar a troca da função hash e redistribuir os valores.



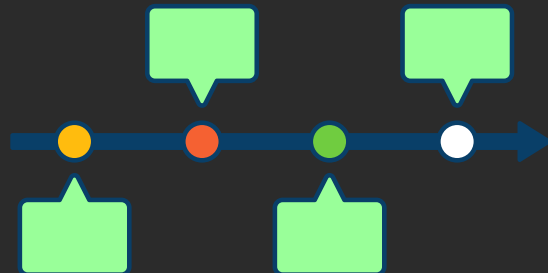
> Exemplo - Encadeamento

0	1	2	3	4	5	6	7	8	9
50			3		15				

Chave	Índice
15	5
23	3
50	0
45	5

> Tratando Colisões

- Endereçamento Aberto
 - Não são utilizadas listas encadeadas ou informações adicionais.
 - Caso haja colisão, é calculado o próximo valor do índice.
 - O índice é calculado até achar uma posição livre ou encontrar a chave (operação de busca).
 - Caso não seja encontrado, a tabela está cheia ou a chave informada não existe.
 - Sondagem Linear ou Sondagem Quadrática.



> Exemplo - Sondagem Linear

0	1	2	3	4	5	6	7	8	9
50			3		15				

Chave	Índice
15	5
23	3
50	0
45	5

> Referências

BOERES, Cristina. ESTRUTURA DE DADOS E ALGORITMOS. **Tabela de Dispersão**, [s. l.], 25 ago. 2019. Disponível em: www2.ic.uff.br/~boeres/slides_ed/ed_TabelaHash.pdf. Acesso em: 16 nov. 2021.

CRUZ, Nelson. **Tabelas de Dispersão**. Hash, [s. l.], 5 maio 2016. Disponível em: www2.unifap.br/furtado/files/2016/11/Aula7.pdf. Acesso em: 16 nov. 2021.

TOFFOLO, Túlio. Algoritmos e Estruturas de Dados I. **Tabelas Hash**, [s. l.], 13 maio 2017. Disponível em: www3.decom.ufop.br/toffolo/site_media/uploads/2013-1/bcc202/slides/23-24._hashing.pdf. Acesso em: 16 nov. 2021.



Obrigado pela atenção!

João Pedro de F. Lourenço
Leandro R. de Souza
Thiago B. Carnaíba
Vinícius A. Marins

joao.franca@aluno.ifsp.edu.br
r.leandro@aluno.ifsp.edu.br
thiago.bruchmann@aluno.ifsp.edu.br
v.marins@aluno.ifsp.edu.br

