Resolução de Problemas Estruturados em Computação Tabela Hash

Bruno Marques

Pontifícia Universidade Católica do Paraná (PUCPR)
Curitiba – PR – Brazil

Resumo. *Nesse artigo será discutido o uso de tabela hash, suas funções e a comparação entre elas, testando o desempenho entre suas variações. O código e os dados analisados nesse relatório se encontram no repositório https://github.com/B0Marques/RA03---Hash*

1. Introdução

É um algoritmo que cria valores hash, sendo em sua maioria números, através de dados recebidos. Seu objetivo é fazer com que os dados sejam distribuídos uniformemente entre a tabela, o que a torna mais eficiente.

Foi escolhido 3 funções hash: **Divisão**, **Multiplicação** e **Dobramento**.

Assim como também foi escolhido cinco tamanhos diferentes para vetores (250, 500, 1000, 5000 e 10000) e cinco tamanho de conjuntos de dados para serem inseridos dentro desses vetores utilizando as funções hash (50000, 100000, 250000, 500000 e 1000000).

O uso da tabela hash vem com vários benefícios, entre eles: a busca rápida, evitar ordenação, a flexibilidade de chaves e entre outros benefícios que fazem o uso de hash ser muito eficaz.

2. Divisão

É uma função fácil de se utilizar, transformando números em chaves através da sobra da divisão (index = Dado resto da divisão por tamanho).

Tamanho 250				
Conjunto de Dados	Tempo de Inserção(ms)	Colisões	Tempo Busca(ns)	Comparações
50000	30	49750	14340	209
100000	137	99750	36240	600
250000	1296	249750	94080	1594
500000	2954	499750	209660	3647
1000000	42018	999750	440360	7673
Tamanho 500				
Conjunto de Dados	Tempo de Inserção(ms)	Colisões	Tempo Busca(ns)	Comparações
50000	12	49500	5600	91
100000	78	99500	17880	297
250000	530	249500	46100	789
500000	5200	499500	104220	1795
1000000	21030	999500	218620	3801
Tamanho 1000				
Conjunto de Dados	Tempo de Inserção(ms)	Colisões	Tempo Busca(ns)	Comparações
50000	7	49000	3740	56
100000	33	99000	9900	159
250000	396	249000	24460	413
500000	1049	499000	53900	908
1000000	17346	999000	111400	1906
Tamanho 5000				
Conjunto de Dados	Tempo de Inserção(ms)	Colisões	Tempo Busca(ns)	Comparações
50000	3	45000	1160	10
100000	10	95000	3620	28
250000	90	245000	7480	77
500000	415	495000	13940	182
1000000	1637	995000	26580	368
Tamanho 10000				
Conjunto de Dados	Tempo de Inserção(ms)	Colisões	Tempo Busca(ns)	Comparações
50000	3	40057	920	4
100000	22	90000	2100	12
250000	112	240000	4260	38
500000	444	490000	8080	85
1000000	990	990000	15000	185

Figure 1. Tabela Divisão

3. Multiplicação

Essa função multiplica o dado por uma constante gerada entre 0 a 1. De tal forma, essa função faz com que a sua tabela hash não possua tanto agrupamento, se escolher a constante de forma certa. Essa função pode proporcionar flexibilidade, ajustando a tabela.

Tamanho 250				
Conjunto de Dados	Tempo de Inserção(ms)	Colisões	Tempo Busca(ns)	Comparações
50000	29	49750	22120	197
100000	112	99750	48820	604
250000	1253	249750	102920	1582
500000	2888	499750	212500	3588
1000000	44556	999750	430480	7543
Tamanho 500				
Conjunto de Dados	Tempo de Inserção(ms)	Colisões	Tempo Busca(ns)	Comparações
50000	11	49500	6780	106
100000	75	99500	18760	306
250000	440	249500	46320	804
500000	4906	499500	100680	1814
1000000	21676	999500	207880	3810
Tamanho 1000				
Conjunto de Dados	Tempo de Inserção(ms)	Colisões	Tempo Busca(ns)	Comparações
50000	6	49000	3840	52
100000	30	99000	10260	152
250000	265	249000	24700	409
500000	807	499000	53680	901
1000000	16380	999000	111200	1908
Tamanho 5000				
Conjunto de Dados	Tempo de Inserção(ms)	Colisões	Tempo Busca(ns)	Comparações
50000	4	45000	1520	10
100000	11	95000	3720	31
250000	86	245000	8220	84
500000	457	495000	14580	183
1000000	1655	995000	26160	386
Tamanho 10000				
Conjunto de Dados	Tempo de Inserção(ms)	Colisões	Tempo Busca(ns)	Comparações
50000	6	40057	1400	4
100000	23	90000	2840	13
250000	94	240000	5640	41
500000	496	490000	9320	89
1000000	1088	990000	21020	185

Figure 2. Tabela Multiplicação

4. Dobramento

Essa função divide o dado recebido como chave em várias partes menores, as somando no final e a dividindo pelo tamanho, criando assim um index com mais agrupamento, se não houver um número certo de dígitos. Funciona bem com chaves que não são apenas números.

Tamanho 250				
Conjunto de Dados	Tempo de Inserção(ms)	Colisões	Tempo Busca(ns)	Comparações
50000	29	49750	14040	205
100000	113	99750	35940	598
250000	1484	249750	91000	1599
500000	3730	499750	199080	3579
1000000	48641	999750	422840	7534
Tamanho 500				
Conjunto de Dados	Tempo de Inserção(ms)	Colisões	Tempo Busca(ns)	Comparações
50000	14	49500	6620	110
100000	88	99500	18120	313
250000	634	249500	44000	802
500000	5631	499500	97300	1813
1000000	24681	999500	203560	3827
Tamanho 1000				
Conjunto de Dados	Tempo de Inserção(ms)	Colisões	Tempo Busca(ns)	Comparações
50000	6	49000	3020	46
100000	31	99000	8800	143
250000	418	249000	23280	411
500000	1914	499000	52560	888
1000000	18842	999000	108860	1891
Tamanho 5000				
Conjunto de Dados	Tempo de Inserção(ms)	Colisões	Tempo Busca(ns)	Comparações
50000	6	45000	1260	12
100000	11	95000	2860	31
250000	166	245000	6220	79
500000	679	495000	13100	177
1000000	1767	995000	26020	386
Tamanho 10000				
Conjunto de Dados	Tempo de Inserção(ms)	Colisões	Tempo Busca(ns)	Comparações
50000	4	40057	840	5
100000	10	90000	2100	17
250000	52	240000	4260	41
500000	312	490000	7860	93
1000000	1087	990000	22680	202

Figure 3. Tabela Dobramento

5. Conclusão

Analisando as tabelas geradas pelo uso das três funções hash, podemos concluir que com a quantidade de dados fornecida, os resultados são muito próximos, apesar da função de divisão ser levemente mais eficaz.

Conforme os números de dados aumentam, as colisões, tempo de inserção e busca aumentam consideravelmente. Entretanto, quanto maior for o tamanho da tabela hash, a tendência é que esses valores diminuam, já que o número de colisões durante a inserção e o número de comparações durante as buscas são menores, demandando menos tempo para a execução.

Apesar dos resultados serem muito similares uns aos outros, é importante pontuar que, em uma situação de maiores quantidades de dados ou diferentes tipos de dados, a eficiência dos diferentes tipos de função se tornam mais evidentes e claros, sendo importante entender as propriedades, vantagens e desvantagens de cada função

específica, garantindo o uso mais apropriado para a tabela em questão, além de como aplicar a função (o ponto flutuante da multiplicação, o tipo de dobramento, etc).