

Trabalho

Tabela Hash

Gustavo dos Santos Capeletti¹, Luis Eduardo Prendim Gontarski², Lucas Borinelli Mees³

¹Engenharia de software - Pontifícia Universidade Católica do Paraná (PUCPR)

Resumo. *Este artigo tem o objetivo mostrar como foram implementadas as funções hash e as classes para o desenvolvimento do Trabalho de Tabela Hash do RA03 - mostrando como foram contruídas as estruturas e também a sua análise de desempenho levando em consideração as métricas tempo de busca, número de comparações, tempo de execução, e o número de colisões.*

1. Classes Utilizadas

Na realização do trabalhos utilizamos 4 classes sendo elas a classe Hash, classe Lista, classe Registro, e a classe Main. Dentro da classe Hash está os métodos de inserir, remover, buscar e imprimir os valores da Tabela. As 3 funções hash estão dentro da classe Hash. Utilizamos a classe Lista para tratar das colisões de valores que possuem o mesmo índice dentro da Tabela. A classe registro é utilizada para criar os dados que serão armazenados dentro da Tabela Hash. Dentro da classe Main é onde os métodos são chamados e onde o código é rodado.

2. Funções Hash

A primeira Função Hash utilizada foi através do módulo da divisão, pegamos o valor do dado que queremos inserir e realizamos o módulo da divisão com o valor do máximo de posições dentro da Tabela Hash.

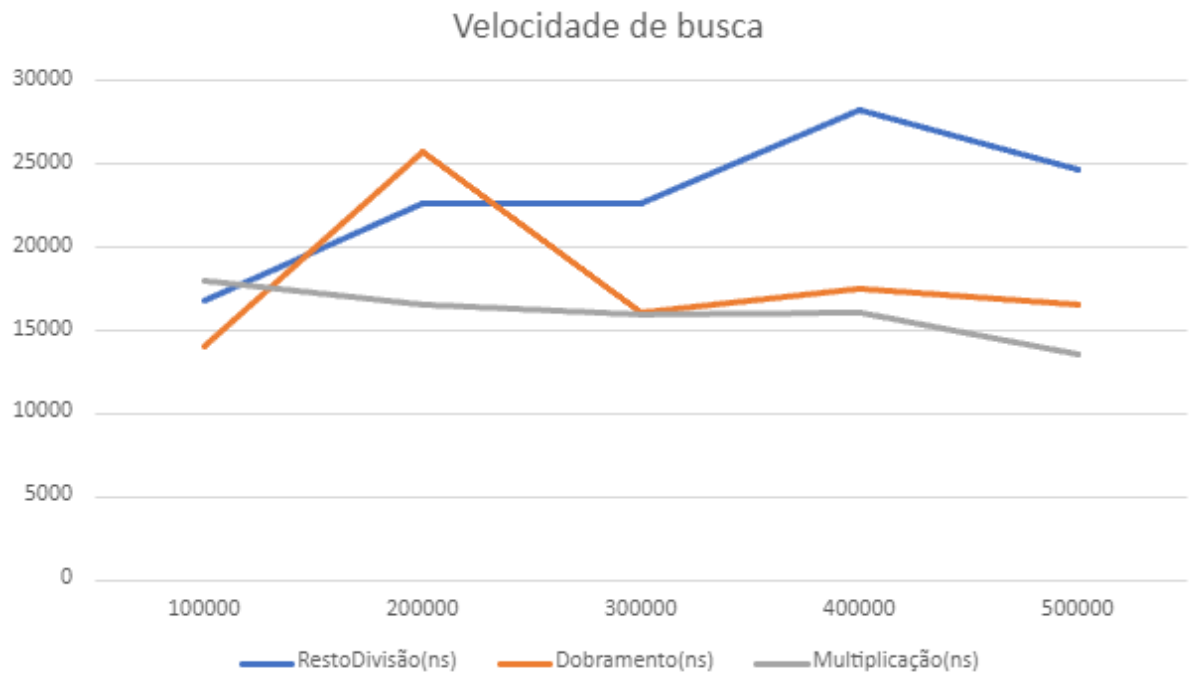
A segunda Função Hash utilizada foi a de Multiplicação, utilizando a função de Multiplicação para converter o índice, com isso é escolhido um número inicial, esse número garantindo uma boa mistura de bits. Ai é multiplicado por 7 também para distribuir os bits de maneira uniforme, assim garantindo de forma mais eficaz uma individualidade do hash.

A terceira Função Hash é a de Dobramento, onde se obtém o valor do chassi, ai é feita uma iteração sobre cada dígito do chassi, ai multiplica a soma por 10, assim criando uma distribuição mais uniforme dos números.

3. Resultados de Velocidade de Busca

Aqui estão os resultados que encontramos analisando as buscas dentro dos vetores dos respectivos tamanhos, 100000, 200000, 300000, 400000 e 500000. E respectivas funções hash, Resto da Divisão, Dobramento e Multiplicação. Os valores se encontram em nanosegundos.

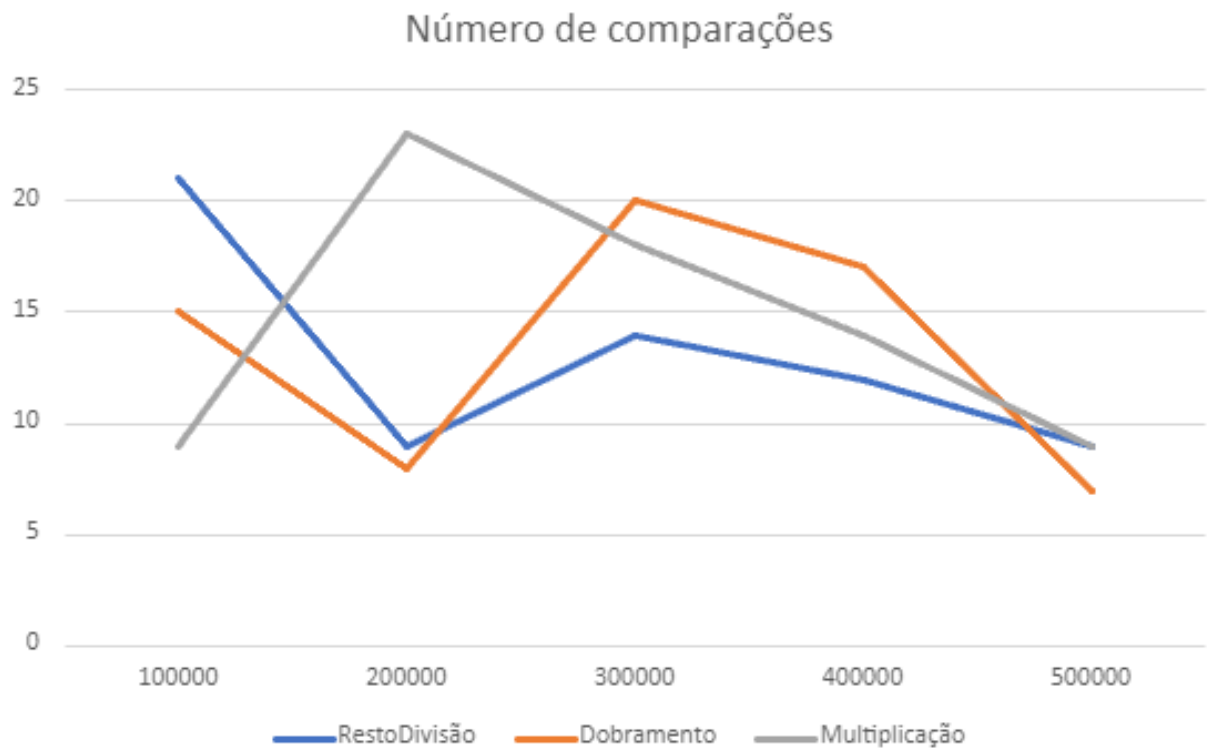
Velocidade de busca							
Tamanho	100000	200000	300000	400000	500000		
RestoDivisão(ns)	16800	22600	22700	28300	24700		
Dobramento(ns)	14000	25800	16100	17500	16500		
Multiplicação(ns)	18000	16500	15900	16100	13600		



4. Resultados de Comparações

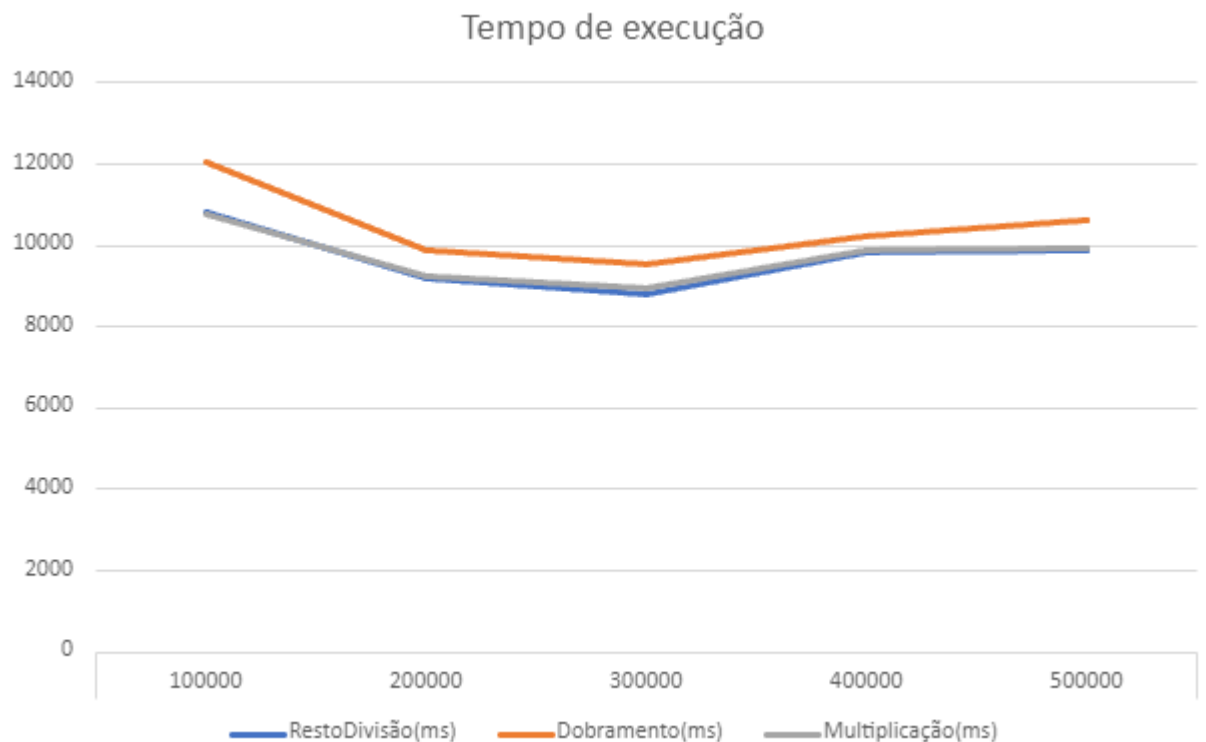
Aqui estão os resultados que encontramos analisando as comparações dentro dos vetores dos respectivos tamanhos, 100000, 200000, 300000, 400000 e 500000. E respectivas funções hash, Resto da Divisão, Dobramento e Multiplicação. Os valores se encontram em números inteiros de comparações que ocorreram dentro do processo de busca.

Comparação							
Tamanho	100000	200000	300000	400000	500000		
RestoDivisão	21	9	14	12	9		
Dobramento	15	8	20	17	7		
Multiplicação	9	23	18	14	9		



5. Resultados de Tempo de Execução

Aqui estão os resultados que encontramos analisando os tempos de execuções dentro dos vetores dos respectivos tamanhos, 100000, 200000, 300000, 400000 e 500000. E respectivas funções hash, Resto da Divisão, Dobramento e Multiplicação. Os valores se encontram milisegundos.

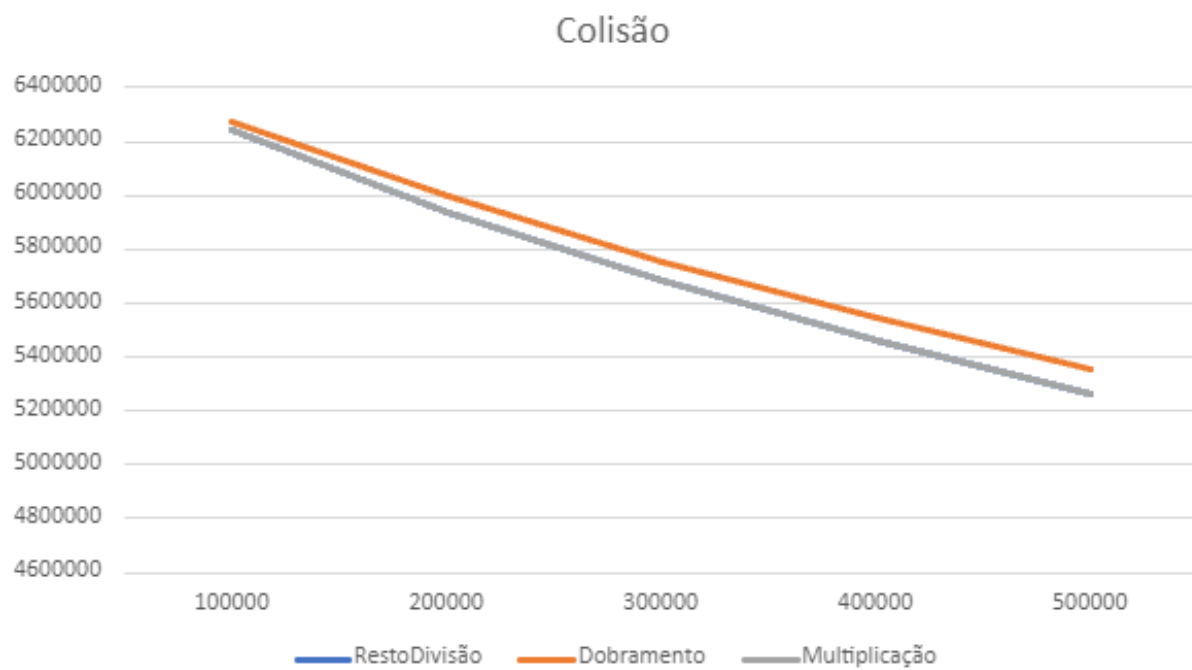


Tempo de execução					
Tamanho	100000	200000	300000	400000	500000
RestoDivisão(ms)	10810	9183	8792	9825	9879
Dobramento(ms)	12038	9890	9530	10222	10627
Multiplicação(ms)	10755	9265	8955	9899	9932

6. Resultados de Colisões

Aqui estão os resultados que encontramos analisando as colisões dentro dos vetores dos respectivos tamanhos, 100000, 200000, 300000, 400000 e 500000. E respectivas funções hash, Resto da Divisão, Dobramento e Multiplicação. Os valores se encontram em números inteiros de colisões que ocorreram durante o processo de inserção.

Colisão					
Tamanho	100000	200000	300000	400000	500000
RestoDivisão	6239392	5939979	5682942	5459345	5261570
Dobramento	6272148	5996391	5756191	5545364	5357540
Multiplicação	6239175	5939989	5683119	5459566	5261293



<https://github.com/IGDSCI/RA3>