

Matheus José da Costa  
11711BCC008

1) O tamanho da tabela hash é definida pelo usuário. Entretanto, a escolha do `tabela_size` pode afetar o desempenho. Por exemplo, é recomendado que esse número seja um número primo, pois isso reduz a probabilidade de colisões. Já um número que é uma potência de dois aumenta a velocidade, mas pode aumentar a probabilidade de colisões se utilizado funções mais simples de hashing. E colisões são, basicamente, dois elementos tentando ocupar o mesmo espaço dentro da tabela.

2) O método de divisão do prof. Backes utiliza a conjunção da chave com a constante `0x7FFFFFFF` para que não haja Overflow. Ou seja, eliminamos o bit de sinal do valor da chave evitando, assim, obter um número negativo.

3) A posição da letra deve ser considerada no cálculo da posição da tabela porque, se eu somar o código de representação em ASCII dos caracteres, independente da ordem a soma dará o mesmo resultado. Por exemplo, tenho duas palavras: cama e maca. Os caracteres são iguais, mas as palavras, não. São semanticamente distintas. Por esse motivo a posição das letras é importante.

Observações: No momento de implementar o remover um elemento de um duplo hash, a minha tabela estava perdendo a minha estrutura de hash. Logo, portanto, não consegui implementá-lo. E, com algumas pesquisas, concluí que realmente não há como fazer a remoção de um duplo hash, justamente pelo problema exposto.