

Exercícios — Tabela Hash

1. Demonstre a inserção das chaves 5, 28, 19, 15, 20, 33, 12, 17 e 10 em uma tabela hash com colisões resolvidas por encadeamento. Considere a tabela com 9 posições e a função hash como sendo $h(k) = k \bmod 9$.
2. Se no tratamento de colisões por encadeamento as listas fossem mantidas de forma ordenada, isso afetaria o tempo de inserção na tabela hash? Por quê? E o tempo de pesquisa seria afetado de alguma forma?
3. Considere uma tabela hash de tamanho $m = 1000$ e a função hash correspondente $h(k) = \lfloor m(kA \% 1000) \rfloor$ para $A = (\sqrt{5} - 1)/2$. Calcule as localizações para as chaves 61, 62, 63, 64 e 65. Obs: Considere $\lfloor \cdot \rfloor$ como arredondamento para baixo.
4. A partir da entrada de um usuário, monte uma tabela hash para armazenar a entrada. Cada palavra pode ser transformada em um número inteiro por meio do somatório do código ASCII de seus caracteres. Estabeleça um tamanho m para a tabela e utilize a função hash: **hash(palavra) = palavra(código ASCII) % m**.
5. Considerando as chaves 4, 5, 7 e 15 e o tamanho do vetor $n = 4$. Utilize a função hash **chave mod n** para alocar estas chaves. Apresente a versão usando o encadeamento e a versão de endereçamento para o tratamento de colisões.
6. Esquematizar o rehashing linear das chaves: 11, 5, 23, 22, 38, 55 e 39. Função inicial: **f(chave) = chave % P**, onde P é o tamanho da tabela.
7. Suponha que estamos utilizando uma Tabela de Hashing com M entradas, com resolução de colisões por rehash linear. Suponha que a função de hashing utilizada seja **$h(chave) = 11k \% M$** , onde chave seja uma letra do alfabeto, K a posição da letra no alfabeto (por exemplo, C corresponde a $k = 3$) e M o número de entradas na tabela. Suponha que desejamos inserir na tabela, inicialmente vazia, as seguintes chaves nesta ordem:

E A S Y Q U T I O N

onde $M = 16$. Ilustre as inserções das chaves na Tabela de Hashing.