

1) Faça um algoritmo de inserção em uma tabela hashing que insere uma chave k em uma tabela T de inteiros utilizando a abordagem de hashing duplo para tratar colisões. Lembre-se de caminhar de forma circular na tabela e de garantir que todas as posições da tabela são testadas uma única vez, ou seja, você precisa criar uma função de hash de colisão que garanta acesso distinto a cada nova colisão (após $i < m$ colisões, i posições distintas de T foram acessadas).

2) Suponha a aplicação do método de endereçamento aberto por tentativa linear, o qual utiliza uma função de dispersão uniforme. Seja um agrupamento primário de tamanho j , isto é, composto por j compartimentos consecutivos ocupados. Mostrar qual a probabilidade do tamanho o agrupamento aumentar para $j+1$ após uma inserção.

4) As versões anteriores do método `hashCode()` do Java para a classe `String` funcionavam sem o uso de todos os caracteres encontrados em cadeias longas. Por exemplo, para uma sequência de dezesseis caracteres, o código de hash foi calculado usando apenas os oito caracteres indexados pares. Explique por que essa foi uma péssima idéia, fornecendo um exemplo de grande conjunto de strings que possuem o mesmo código de hash.