Questões teóricas interessantes sobre Tabelas Hash:

1- Quais são as características de uma boa função de hash?

2- Qual a característica de uma boa chave?

3- Qual seria o impacto causado por uma função de hash aleatória? Isto é, para uma mesma chave ela pode retornar diferentes valores de hash a cada chamada.

4- Tabelas Hash são ordenadas? Dados dois objetos, é possível saber qual elemento foi adicionado antes de outro?

5- Se eu quiser saber se dois arquivos de texto são idênticos, basta apenas comparar o hash dos dois. Como fazer isso em sistemas unix?

6- Qual o tamanho inicial de um HashMap em Java?

7- Qual o fator de carga limite para que seja feito um rehash em Java?

Respostas:

1- Quanto maior for a dificuldade de se criar colisões intencionais, melhor é a função hash. Para isso a função precisa ser unidirecional e usar mod por números primos.

Mais Características:

Unidirecionalidade: conhecido um resumo h(M), deve ser computacionalmente impossível encontrar M a partir do resumo.

Compressão: a partir de uma mensagem de qualquer longitude, o resumo h(M) deve ter uma longitude fixa. O normal é que a longitude de h(M) seja menor do que a da mensagem M.

Facilidade de cálculo: deve ser fácil calcular h(M) a partir de uma mensagem M.

Difusão: o resumo h(M) deve ser una função complexa de todos os bits da mensagem M: se, se modifica um só bit da mensagem M, o hash h(M) deveria mudar a metade dos seus bits aproximadamente.

Colisão simples: será computacionalmente impossível, conhecido M, encontrar outro M’ tal que h(M) = h(M’). Isto se conhece como resistência débil às colisões. Colisão forte: será computacionalmente difícil encontrar um par (M, M’) de forma que h(M) = h(M’). Isto se conhece como resistência forte às colisões.

2- A chave é o retorno de uma função hash, para isso ser bom é preciso que seja um valor fixo.

3-

4- Pode inserir elementos já ordenados. Não preserva a ordem das chaves, sendo assim não podemos saber quem foi adicionado primeiro do que outro.

5- É preciso gerar hashCodes para os arquivos, salvar e depois verificar se são iguais:

**gerar hashCode de um arquivo e salvar: md5sum nome\_do\_arquivo > nome\_do\_arquivo.txt**

Por fim, abre os dois arquivos gerados e verifica se são iguais.

6- Tamanho default de um HashMap em Java é 16.

7- Fator de carga limite em Java é 0.75(75%), quando atinge isso será feito o reHash.