**Filtros de Bloom**

FONTE: http://codekata.com/kata/kata05-bloom-filters/

Em muitas circunstâncias precisamos descobrir se um determinado dado é membro de um conjunto, e existem muitos algoritmos para fazê-lo. Se o conjunto é pequeno, você pode usar bitmaps. Quando os conjuntos se tornam maiores, *hashes* são uma técnica útil. Mas quando os conjuntos ficam realmente grandes, começamos a bater em limitações. Por exemplo, manter 250.000 palavras em memória para um corretor ortográfico pode exigir um espaço de memória muito grande em um telefone celular.

Filtros de Bloom são uma técnica estatística desenvolvida a mais de 30 anos para testar se um elemento pertence a um conjunto. Eles reduzem a quantidade de armazenamento necessário para representar o conjunto, mas a um preço: às vezes os filtros relatam que um determinado dado pertence ao conjunto, quando não pertence (mas nunca o oposto, ou seja, se o filtro diz que o conjunto não contém o seu dado, você sabe que ele realmente não pertence ao conjunto). Você ainda pode controlar a precisão dos filtros de Bloom: quanto mais memória você estiver preparado para investir no algoritmo, menor será o número de falsos positivos.

Filtros de Bloom são muito simples. Comece com uma grande matriz de bits, inicialmente todos com valor igual a zero. Em seguida, considere os dados que você quer procurar no conjunto (no nosso caso, um dicionário de palavras). Gere *N* valores de hash independentes para cada palavra. Cada hash é um número usado para definir o bit correspondente na matriz, que deverá ser ligado (valor igual a 1). Às vezes haverá confrontos, situações em que o bit já está associado a outra palavra. Isso não importa.

Para verificar se uma palavra está no dicionário, aplique nela os mesmos *hashes* usados para criar o bitmap. Em seguida, verifique se cada um dos bits correspondentes a estes hashes está ligado. Se algum bit não estiver definido, você não carregou inicialmente essa palavra e pode rejeitá-la.

O filtro de Bloom relata um falso positivo quando todo o conjunto de hashes de uma palavra aponta bits que foram ligados por outras palavras. Na prática, isso não acontece com muita frequência, a não ser que a matriz esteja muito sobrecarregada com bit. Há uma discussão sobre a matemática por trás dos filtros de Bloom em www.cs.wisc.edu/~cao/papers/summary-cache/node8.html.

Este exercício consiste em implementar um corretor ortográfico baseado filtro de Bloom. Você vai precisar de algum tipo de matriz de bits, algumas funções de hash, uma maneira simples de ler as palavras componentes do dicionário e, finalmente, as palavras que deseja verificar. Para a função hash, você pode usar um algoritmo que gere um *hash* longo (como o MD5) e pegar partes do resultado gerado para ligar os bits.