

A operação de remoção

Da maneira como foi definido acima, o filtro de Bloom não suporta a operação de remoção.

Porque ao remover um elemento da base de dados, nós não podemos simplesmente ajustar os bits correspondentes no filtro para 0 — (*porque?*)

Mas uma pequena esperteza resolve esse problema.

Quer dizer, a ideia é trocar os bits por contadores

F	0	2	0	0	1	1	3	0	1	2	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---

onde cada contador indica quantos elementos na base de dados selecionaram essa posição.

Daí, na operação de inserção os contadores são incrementados.

Na operação de remoção os contadores são decrementados.

E o filtro responde NÃO para uma consulta quando ao menos um dos contadores tem valor 0.

O problema com essa solução é que os contadores não podem ocupar muito espaço

— (*para que o filtro não ocupe muito espaço ...*)

Quer dizer, a ideia é que os contadores tenham apenas 2 ou 3 bits.

Mas um contador com 3 bits só pode contar até 7.

E quando ele chega em 7, novos incrementos mantém o seu valor em 7.

Mas daí, nós já não podemos mais decrementar o valor do contador.

Porque nós não sabemos se existem 7 ou mais que 7 elementos na base de dados que selecionam aquela posição.

Quer dizer, os contadores que alcançam o valor máximo não podem mais ser decrementados.

E nós vemos que essa não é uma solução perfeita para o problema da remoção.