

Cuckoo Hash

Vinicius Tikara Venturi Date

27 de setembro de 2024

1 Objetivo

O objetivo do trabalho é a implementação de uma versão simplificada do *Cuckoo Hashing*. Nesta versão não temos o *rehashing*, apenas a utilização de duas funções hash para a busca, inserção e exclusão de elementos..

As duas funções de hash são consideradas, $h1(k) = k \bmod m$ e $h2(k) = m \cdot \lfloor (k \cdot 0.9 - \lfloor k \cdot 0.9 \rfloor) \rfloor$, com $\lfloor x \rfloor$ denotando piso de x .

Três funções são estudadas para a realização do algoritmo: A primeira delas é a *busca*, que consiste em procurar a chave k nas tabelas. Começamos pela tabela T1 utilizando sua função de hash $h1(k)$ para o endereçamento. Caso o endereço esteja vazio, retornamos essa informação. Se não for o caso, então, se a chave estiver em T1, retornamos a posição dela em T1. Por fim, se não estiver em T1, então a chave é procurada em T2, utilizando a função hash $h2(k)$ para calcular sua posição, e caso esteja lá, é retornado o endereço calculado.

A segunda função é relacionada à inserção das chaves. O algoritmo primeiro tenta inserir em a chave em T1. Existem dois casos, caso o valor não exista em T1, podemos inserir e retornar. Caso contrário, precisamos procurar uma entrada na segunda tabela T2, copiar o valor em T1 para T2 nessa nova entrada, e inserir o dado em T1.

Por fim, a última função consiste na exclusão das chaves das tabelas T1 e T2. A função utiliza as funções hash apropriadas, primeiro testando pela sua existência em T2, caso seja encontrada lá, excluimos essa entrada. Caso contrário procuramos pela chave em T1, tomando certo cuidado, pois existem chaves em T2 que dependem de chaves em T1. Precisamos manter essa informação enquanto excluimos os valores de T1, essa informação será guardada pelo estado "excluído" dos elementos.

2 Implementação

A implementação foi feita em C++, utilizando-se das suas estruturas de dados para a realização das três funções pedidas. São montadas três structs:

A primeira estrutura representa um elemento de qualquer tabela, chamada de `elemento` ela contém o valor da chave e suas duas flags de estado: vazia e excluída.

A segunda estrutura de nome `tabela_hash` representa ambas tabelas hash. Essa estrutura é implementada com um *vector* de elementos.

Por fim, a última estrutura é o `cuckoo_hash`, que é implementado com duas tabelas hash e um valor `m`, que representa o tamanho de cada uma das tabelas.

As funções são implementadas utilizando a descrição da seção "Objetivo".

3 Conclusão

O estudo do Cuckoo Hash aliado a sua implementação nos permitem entender melhor como os algoritmos de busca, inserção e remoção de chaves ocorrem nesse tipo de estrutura, e também os custos muitas vezes escondidos pelas chamadas de funções já prontas, sejam do sistema ou implementadas por outros desenvolvedores.