Universidade Federal de Lavras

Amorim, Castro e Lima

Estruturas de Dados – GCC 216

Sistema de cadastro em Hashing Extensível

Lavras

2018

Davi Horner Hoe de Castro

Gabriel Henrique Silva Amorim

Thiago Luigi Gonçalves Lima

Estrutura de Dados – GCC 216

Trabalho apresentado junto ao curso de Ciência da Computação da Universidade Federal de Lavras, como requisito parcial para obtenção de nota.

Orientadores: Profs. Drs. Joaquim Quinteiro Uchôa e Juliana Galvani Greghi

Lavras

2018

Introdução

Nesta terceira etapa do projeto nos foi instruído que juntassemos a primeira parte com a segunda. Então faríamos o sistema de cadastro com hashing extensível utilizando a ideia da árvore *trie.*

Código

Como na segunda etapa já foi abordado o tema de árvore *trie* para esta terceira bastou apenas adaptar o mesmo para o novo cenário de uso. Não seria mais usado para palavras e sim para números binários.

Métodos

**insereHash**

Modificamos este método para adicionar alguns métodos novos que foram criados para adequar o código ao uso da hash extensível.

**redimensionarHash**

Este método foi criado para redimensionar a hash quando um bloco estiver cheio. Tornando-a uma verdadeira hash extensível.

**pegarDados**

Pega a posição da hash para ser utilizada.

**testarCompChave**

Verifica se a chave mandada é a mesma que a chave do bloco.

**resetaBloco**

O reseta foi criado pois não tinha como inserir no bloco cheio quando redimensionasse.

**reinsereBloco**

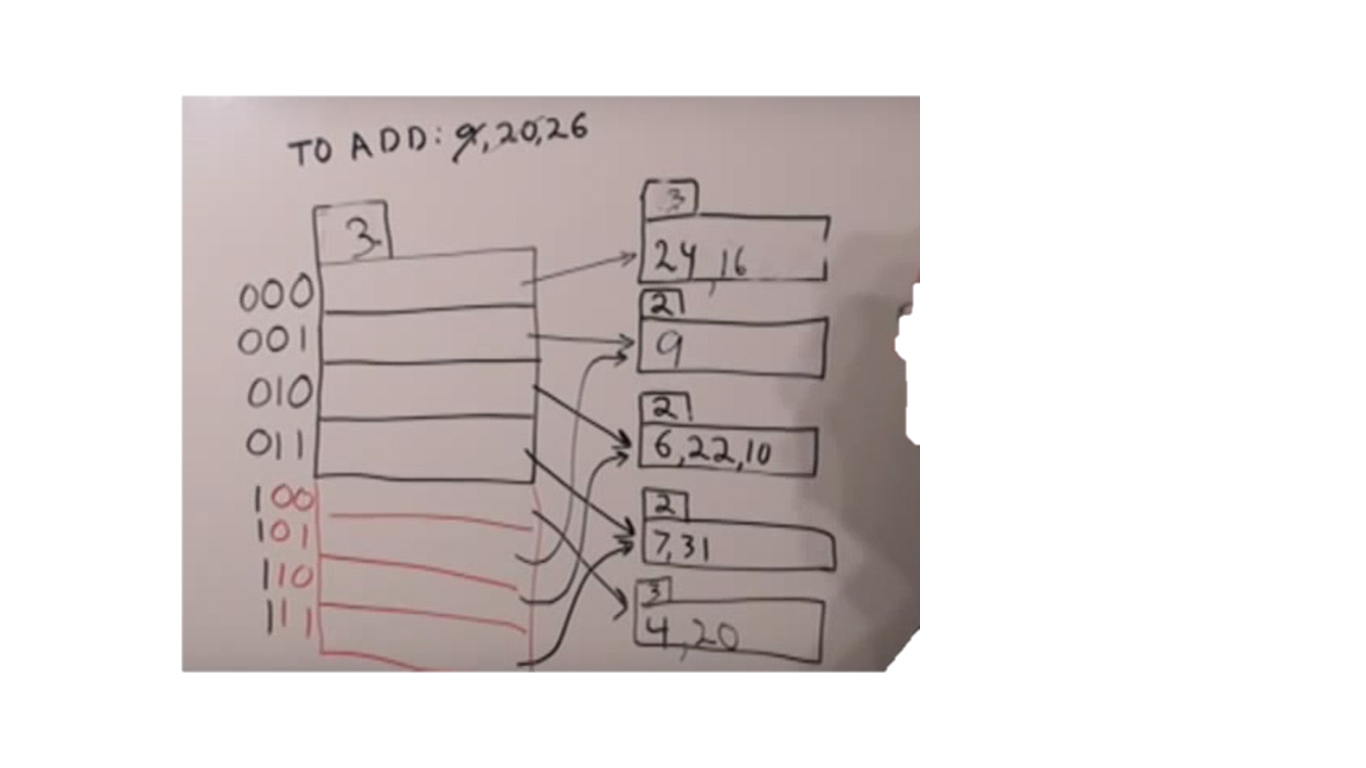
O reinsere só fica chamando a insere para colocar de novo os dados do bloco resetado.

**pegarCompChave**

O pegarCompChave é para pegar o compChave do bloco para atualizar.

Funcionamento do Programa

Inicialmente ele começa como um hash de tamanho 2 e bloco de tamanho 4. Nas primeiras cinco inserções todos os dados são inseridos no bloco apontado pela primeira hash. Na quinta inserção ocorre a quebra do bloco e a hash dobra de tamanho e os valores são reinseridos na hash, como toda posição da hash tem o valor da posição dela em binário quando ela é dobrada passa a se olhar uma casa a menos no valor do binário, a inserção continua igual, caso exista alguma posição da hash vazia ela irá apontar para o seu igual na parte de cima.



(Imagem ilustrando o funcionamento/Imagem meramente ilustrativa)

Agora se espera a nova quebra do bloco para a hash ser duplicada novamente e o processo se repetir.

Conclusão

Foram várias horas e esboços tentando entender este conteúdo que é tão abstrato. Essas estruturas possuem uma matemática e, principalmente, uma lógica bem pesada por trás.

O trabalho como um todo foi extremamente difícil de ser implementado. Porém, a experiência adquirida com o mesmo foi incrível. Vimos na prática a utilização das estruturas de dados em um sistema de cadastro que, embora, agora esteja armazenando poucos dados (por escolha do grupo) é escalável e poderia ser usado para um mini banco de dados.