

TAD HASH (Linear Probing)

Daniel de Souza Rodrigues - 18.2.8112 - Sistemas de Informação

¹Instituto de Ciências Exatas e Aplicadas – Universidade Federal de Ouro Preto (UFOP)
Rua 36, Número 115 - Bairro Loanda, João Monlevade - CEP: 35931-008

daniel.sr@aluno.ufop.edu.br

Resumo. *Este artigo aborda o tema (Linear Probing) usado para solucionar conflitos de chaves em uma tabela hash; busca por elemento e remoção de elementos da tabela hash são abordados brevemente com possíveis soluções e implementações.*

1. Introdução

O trabalho implementa o (Linear Probing) como maneira de resolução para os conflitos de chaves(key) em uma tabela hash, ou seja chaves que possuem o mesmo valor quando é executado a função hash, assim quando as chaves são iguais um novo local para armazenamento é buscado na hash caso seja encontrado o valor é armazenado na posição vazia seguinte, também foi implementado a função de busca por elemento e a função delete a qual remove o elemento da tabela hash mas não remove sua memória alocada, tornando a posição do elemento removido uma posição livre a qual pode receber novos valores.

2. Implementação

2.1. Search-miss

Foi implementada a função search-miss de maneira que a função recebe como parâmetro uma chave(key), logo a função executa um laço onde a variável “i” recebe a hash(key) e percorre toda a tabela comparando se a tabela keys[i] é igual o valor da chave recebido como parâmetro, caso seja igual a função retorna o valor contido na variável “i”, caso não seja satisfeita a condição retorna (-1) .

2.2. Delete

A função delete recebe como parâmetro uma key, a função possui uma variável “aux” do tipo int, essa variável recebe o valor mod concedido pela função count-search-miss(key), logo a função compara se aux é diferente de empty, caso for a função executa um laço começando em $j = aux + 1$ e percorre toda a tabela hash executando uma comparação com o mod de cada valor posterior a “aux” e comparando com o valor aux, caso a condição seja satisfeita $vals[aux] = vals[j]$ e $keys[j]$ recebe empty, ou seja, caso exista um valor que possui o mesmo mod na função ele é copiado para uma nova posição, mantendo a condição da chave para não atrapalhar nas futuras buscas de outros elementos que possuam a mesma key, caso exista apenas um valor que possui a chave key, a tabela executa $keys[aux]$ e $vals[aux]$ atribuindo empty nesta determinada posição, caso nenhuma condição seja efetuada a função retorna -1.

2.3. Alfa

Foi implementado uma função alfa a qual calcula o alfa típico baseando se nas formulas derivadas por Knuth em 1962.

3. Custo Médio

Foram realizados testes com hash de tamanho 1.000, 10.000, 100.000, 1.000.000 com inserção (PUT) randomica, todas execuções validão as equações propostas por formulas derivadas por Knuth em 1962. Fica evidente o desperdício de memória principalmente em casos onde o tamanho da hash é superior a 100, no caso do valor 1.000.000 mais da 70% da memória é alocada desnecessariamente e existe uma enorme aglomeração de chaves no centro da hash.[Thorup 2009]



Figure 1. Detecção de conflitos em Hash (Linear Probing)

3.1. Se M (tamanho da tabela hash) é muito grande. O que acontece?

Se M é alto, muitas posições ficam vazias no array.

3.2. Se M (tamanho da tabela hash) é muito pequena. O que acontece?

Se M é pequeno, o tempo de busca(search) aumenta devido ao agrupamento.

3.3. Qual é o $\alpha = N/M = ?$ típico? Por quê?

- Para $\alpha = 1/2$, sondagens para busca é cerca de $3/2$.
- Para $\alpha = 1/2$, sondagens para inserção é cerca de $5/2$.

3.4. Delete

Com esta implementação utilizada na função delete problema exibido de buscarmos H que possui o mesmo hash de C na tabela é resolvido, pois apenas é removido o primeiro valor buscado como o parâmetro (key), no caso C. [Wikipedia contributors 2019]

4. Dificuldades na implementação

Encontrei muitas dificuldades para entender, implementar e analisar o hash, enviei um e-mail visando sanar algumas dúvidas mas não obtive resposta.

References

Thorup, M. (2009). String hashing for linear probing. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 655–664, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Wikipedia contributors (2019). Linear probing — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Linear_probing&oldid=888756393. [Online; accessed 8-July-2019].