

Projeto I - Tabela DNS com Skip Lists

Victor Henrique de Souza Rodrigues - 9791027

¹Instituto de Ciências Matemáticas e de Computação - ICMC

victor.henrique.rodrigues@usp.br

Resumo. Este documento tem por finalidade mostrar a diferença de execução e de implementação entre dois TADs distintos, Skip List e Lista Ligada Simples.

1. Lista Ligada Simples

A Lista Ligada é uma estrutura dinâmica e linear, onde os elementos são inseridos em nós que contém um ponteiro para o próximo elemento da lista.

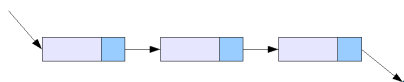


Figura 1. Exemplo de Lista Ligada Simples

1.1. Vantagens da Lista Ligada Simples

- A Inserção não altera outros elementos da lista, uma vez que usualmente é feita no começo da lista
- Pelo fato de ser dinâmica, permite a utilização sem alocação de memórias em excesso (desnecessárias)

1.2. Desvantagens da Lista Ligada Simples

- Para procurar um elemento é preciso percorrer a lista até encontrá-lo

1.3. Complexidade Lista Ligada

Operações	Complexidade (pior caso)
Inserção	$O(1)$
Remoção	$O(n)$
Busca	$O(n)$

As informações da tabela acima são referentes a minha implementação

1.4. Implementação em C

1.4.1. Inserção

Para a inserção é preciso alocar um novo nó, e fazer o próximo elemento desse novo nó receber o primeiro e o primeiro receber esse novo nó.

1.4.2. Remoção

Para a remoção é preciso percorrer a lista até encontrar o elemento que deseja ser removido, fazer o ponteiro próximo do anterior apontar para o próximo do atual, e liberar o atual.

1.4.3. Pesquisa

Para a pesquisa é necessário percorrer a lista se encontrar o elemento, retorna o mesmo, caso chegue ao final e não encontre, retorne -1.

2. Skip List

A Skip List é uma estrutura que combina elementos de árvore binária com lista ligada, uma vez que dependendo de uma distribuição uniforme de probabilidade o elemento pode ser inserido também um nível acima.

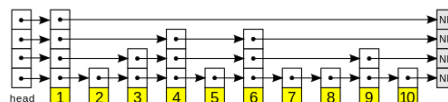


Figura 2. Exemplo de Skip List

2.1. Vantagens da Skip List

- Com a inserção em níveis, a busca se torna muito mais eficiente, semelhante a uma busca binária, uma vez que não precisa percorrer a lista inteira para retornar o elemento.

2.2. Desvantagens da Skip List

- Por ser uma implementação recente, há pouco material disponível sobre ela
- Dependendo da linguagem a implementação não é tão simples
- Ocupa mais memória que uma lista ligada simples

2.3. Complexidade Skip List

Operações	Complexidade
Inserção	$O(\log n)$
Remoção	$O(\log n)$
Busca	$O(\log n)$

As informações da tabela acima são referentes a minha implementação

OBS: Uma vez que é um algoritmo probabilístico, a chance de cair no pior caso, onde todas as operações seriam em $O(n)$, para um número grande de entradas, tente a zero.

2.4. Implementação em C

2.4.1. Inserção

Para a inserção, é necessário parar um ponteiro em um elemento antes da qual deseja inserir, criar um novo nó, fazer o ponteiro anterior apontar para ele, e o seu ponteiro apontar para o próximo, além de "sortear" e a cada SIM inserir no nível acima, até que um NÃO seja "sorteado para terminar a inserção

2.4.2. Remoção

Para a remoção é preciso chegar ao elemento que deseja ser removido, fazer o ponteiro anterior apontar para o próximo elemento depois dele, e remover ele e todos os nós iguais a ele acima

2.4.3. Pesquisa

Para a pesquisa, é necessário percorrer alguns elementos, em uma espécie de busca em profundidade misturada com busca sequencial, até chegar no elemento desejado e retorná-lo.

3. Conclusão

A Skip List se torna uma ótima alternativa a lista ligada simples para entradas muito grandes, uma vez que graças a sua estrutura (semelhante a estações de trem, que mistura lista ligada com árvores de busca) as principais operações se tornam de ordem logarítmica $O(\log n)$, apesar de ser pior na inserção do que uma lista ligada a sua busca e sua remoção fazem a sua implementação ser mais vantajosa do que uma lista ligada simples, para um número grande de elementos.