

Atividade 10 - Prática

Entrega: 09/02/2020

Análise de complexidade do código para Tabelas Hash sem colisões e de endereçamento aberto.

A estrutura da Tabela Hash é dada de maneira muito semelhante a um vetor alocado de maneira dinâmica. A principal diferença de uma Tabela Hash é a inserção de chaves para a posição dos valores dentro da tabela. Além da função de hash, temos também métodos para melhor distribuir os valores nas chaves, que são dadas por meio de um endereçamento aberto.

	Tempo de Inserção	Buscar o Valor 999
Hash por Encadeamento	0.001047 segundos	0.000008 segundos
Hash por Endereçamento Aberto	0.008045 segundos	0.000015 segundos

Figura 1 - Tabela dos Tempos de cada uma das ações pedidas.

```
O algoritmo de insercao na Tabela Hash demorou 0.001047 segundos
O numero 999 foi encontrado!!
O algoritmo de busca na Tabela Hash demorou 0.000008 segundos
O algoritmo de insercao na Tabela Hash de Endereçamento Aberto demorou 0.008045 segundos
O numero 999 foi encontrado!!
O algoritmo de busca na Tabela Hash de Endereçamento Aberto demorou 0.000015 segundos
```

Figura 2 - Os Outputs dos Hashs para Busca e para a Inserção.

Analisando a complexidade de cada uma das ações, temos que as complexidades das estruturas de dados de Hash são dadas, na inserção por $O(n)$ no pior dos casos, pois terá de percorrer toda tabela para encontrar a chave correta, ou seja, será numa complexidade linear de acordo com o vetor de inserção. Entretanto, a busca é dada de maneira constante, ou seja, $O(1+c)$, sendo c a quantidade de operações para buscar o valor. Como a constante c pode ser ignorada, temos que a complexidade é dada por $O(1)$.