

Atividade – Tabela Hash

Implemente as atividades descritas abaixo.

Essa implementação pode ser feito em grupo de até 3 pessoas.

Envei todos os códigos desenvolvidos para o STOA.

Não se esqueça de colocar nos códigos os nomes de todos os integrantes do grupo.

Data de Entrega: **22/11/2017** até as **22:50** via STOA.

Atividade 1: Inserção em Tabela Hash

Implemente em C um conjunto de funções para inserção de dados em uma Tabela Hash. A função de espalhamento da hash será a função mod (%) em relação ao tamanho da tabela. Dessa forma, um elemento x deverá ser inserido na posição $x \% M$, sendo M o tamanho da hash. Você deverá implementar a abordagem de lista encadeada para tratar colisões.

Nessa implementação você deverá criar uma lista encadeada para cada uma das posições da tabela Hash. Ao inserir um elemento na hash você deverá inserir esse elemento no início da lista encadeada, isso irá garantir que a operação de inserção executará em $O(1)$.

Nessa atividade iremos inserir apenas números inteiros, por isso a função mod poderá ser facilmente utilizada. Faça testes com $M = 10$ inicialmente, assim facilita o debug dos códigos. Em seguida utilize o valor de $M = 9$, afim de melhorar a sua função de espalhamento.

Exemplo de valores inseridos: 5, 2, 3, 8, 10, 14, 28, 21, 24, 19 na tabela hash com $M = 9$.

Saída esperada para esse exemplo de entrada:

```
id -> chave
1 -> 19, 28, 10
2 -> 2
3 -> 21, 3
5 -> 14, 5
6 -> 24
8 -> 8
```

Atividade 2: Criação de Índice Remissivo

Adapte o código anterior desenvolvido em C para o contexto descrito abaixo:

Problema de criação de índice remissivo: Várias aplicações necessitam de um relatório de referências cruzadas. Por exemplo, a maioria dos livros apresentam um índice remissivo que corresponde a uma lista alfabética de palavras chave ou palavras relevantes do texto com a indicação dos locais no texto onde cada palavra chave ocorre.

Como exemplo, suponha um arquivo contendo um texto constituído como abaixo:

```
Linha 1: Good programming is not learned from  
Linha 2: generalities but by seeing how significant  
Linha 3: programs can be made clean easy to  
Linha 4: read easy to maintain and modify  
Linha 5: human-engineered efficient and reliable  
Linha 6: by the application of common sense and  
Linha 7: by the use of good programming practices
```

Assumindo que o índice remissivo seja constituído das palavras chave:

and, by, be, easy, human-engineered, programming, programs, to

o programa para criação do índice deve produzir a seguinte saída:

```
and -> 6, 5, 4  
be -> 3  
by -> 7, 6, 2  
easy -> 4, 3  
human-engineered -> 5  
programming -> 7, 1  
programs -> 3  
to -> 4, 3
```

Note que a lista de palavras chave está em ordem alfabética. Adjacente a cada palavra chave está uma lista de números de linhas, um para cada vez que a palavra ocorre no texto.

Projete um sistema para produzir um índice remissivo como projetado acima. Adapte a função hash a esse novo contexto. Como nesse exemplo temos apenas 8 palavras chaves crie uma tabela hash com 8 posições.