

## INF 213 - Roteiro da Aula Prática

Objetivo: praticar (mais) o uso de árvores binárias de pesquisa e ABPs com balanceamento AVL.

→ LEMBREM-SE DE USAR PAPEL E CANETA COMO RASCUNHO ANTES DE IMPLEMENTAR <<--

Arquivos fonte e diagramas utilizados nesta aula:

[https://drive.google.com/open?id=1qtsoUh\\_hjx881dtbcDM7keRFfHzSXR8g](https://drive.google.com/open?id=1qtsoUh_hjx881dtbcDM7keRFfHzSXR8g)

### Etapa 1

Mostre em uma folha de papel cada passo da criação de uma árvore AVL armazenando inteiros (indique o fator de balanceamento de cada nó em cada etapa) supondo que os números serão inseridos na seguinte ordem:

7,0,3,5,4,2,1,8,6,9

Para entregar esta etapa, mostre seus desenhos para o professor. (no caso do ensino remoto, esta etapa não precisará ser entregue/mostrada -- mas poderá ser cobrada nas entrevistas).

### Etapa 2

Considere o arquivo MySet.h fornecido para a aula. Termine a implementação do operador de incremento da classe do iterador. Teste sua implementação utilizando o programa testaMySet.cpp (estude a implementação do testaMySet.cpp). Lembre-se de compilar o programa principal com a flag -std=c++11 (visto que estamos usando o “range-based” for do C++11).

### Etapa 3

Considere o arquivo testaMySetTipoCustomizavel.cpp fornecido para a aula. Termine a implementação da classe Aluno de modo que ela possa ser armazenada na nossa estrutura de dados MySet.

Faca essa implementação de modo que, ao percorrer a estrutura de dados utilizando iteradores, alunos com nomes lexicograficamente maiores apareçam primeiro (ou seja, Zeze apareceria antes de Capivaristo). Se dois alunos possuírem o mesmo nome desempate pela matrícula (matrículas maiores deverão aparecer antes de matrículas menores).

Obs: não modifique a classe MySet! Modifique apenas a classe Aluno!!!

### Etapa 4

Crie um programa chamado “imprimeProximo.cpp”. Seu programa deverá fornecer duas funcionalidades: cadastrar números inteiros em um “banco de dados” e imprimir o sucessor de um determinado número (do banco de dados).

Ao cadastrar um numero n (isso sera representado por uma linha na entrada contendo o caractere C seguido do numero n), n deve ser inserido no “banco de dados” (o banco de dados não devera armazenar valores repetidos).

Se a operacao sucessor de um numero n for requerida (isso sera representado por uma linha na entrada contendo o caractere S seguido do numero n), seu programa devera imprimir uma linha contendo o menor numero maior que n que estiver no banco de dados no momento. Se n não estiver cadastrado ou se n não tiver um sucessor cadastrado, imprima uma linha contendo a palavra “FALHA”.

Observe que isso poderia ser implementado facilmente utilizando um vetor dinamico para armazenar os elementos. Porem, ou a operacao de insercao seria muito lenta ou a operacao de encontrar sucessor seria lenta (por que?). Assim, use uma estrutura de dados eficiente para realizar esse processamento de forma rapida.

Entrada	Saida esperada (a explicacao entre parenteses não estara na saida!)
C 1	5 (o banco de dados possui 1,4,5,10)
C 5	FALHA (9 não está cadastrado ainda...)
C 10	FALHA (10 não possui sucessor cadastrado)
C 4	10 (o banco de dados possui 1,4,5,9,10)
C 1	4 (o banco de dados possui 1,4,5,9,10)
S 4	
S 9	
S 10	
C 9	
S 9	
S 1	

### Submissao da aula pratica:

A solucao deve ser submetida ate as 18 horas da proxima Segunda-Feira utilizando o sistema submittty ([submittty.dpi.ufv.br](http://submittty.dpi.ufv.br)). Envie todos os arquivos fonte (tanto os arquivos .h e .cpp fornecidos neste laboratorio quanto os que você implementou). Atualmente a submissao so pode ser realizada dentro da rede da UFV.