

Universidade Federal do Ceará - Campus Quixadá  
QXD0115 – Estrutura de Dados Avançada – Turma 02A  
Curso de Ciência da Computação  
Prof. Atílio Gomes Luiz

Aplicação de Árvores AVL em Conjuntos Dinâmicos
---

1. Um **conjunto** é uma coleção de elementos. TODOS os elementos de um conjunto são diferentes entre si. Um exemplo de conjunto muito estudado é o conjunto dos números inteiros  $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$ . Um outro exemplo é o conjunto dos nomes de pessoas que começam com a letra A, por exemplo {Ana, Alice, Atílio, Alinson, Alex, ...}.

Em computação, os conjuntos que são manipulados por algoritmos podem aumentar e diminuir de tamanho, ou seja, podem mudar com o passar do tempo. Tais conjuntos são chamados **conjuntos dinâmicos**. Para esta atividade, deve ser desenvolvida uma estrutura de dados, chamada SET, que implementa um conjunto dinâmico  $S$  fornecendo pelo menos as operações básicas listadas a seguir:

- CONSTRUTOR: método especial que cria um conjunto vazio.
- DESTRUTOR: método especial que é invocado quando o conjunto é destruído. Ele deve liberar qualquer recurso que foi alocado dinamicamente durante o uso da estrutura de dados.
- INSERT( $x$ ): insere o inteiro  $x$  no conjunto  $S$ .
- ERASE( $x$ ): remove o inteiro  $x$  do conjunto  $S$ , se ele estiver presente.
- CONTAINS( $x$ ): devolve TRUE ou FALSE indicando se  $x$  está ou não no conjunto  $S$ , respectivamente.
- CLEAR(): deixa o conjunto vazio.
- SWAP( $T$ ): recebe como entrada um conjunto  $T$  e troca o conteúdo de  $T$  com o de  $S$ .
- MINIMUM(): devolve o valor do menor elemento contido no conjunto  $S$ .
- MAXIMUM(): devolve o valor do maior elemento contido no conjunto  $S$ .
- SUCCESSOR( $x$ ): essa operação deve retornar o valor do sucessor de  $x$  no conjunto  $S$  ou pode lançar uma exceção caso  $x$  não tenha sucessor ou caso  $x$  não esteja no conjunto.
- PREDECESSOR( $x$ ): essa operação deve retornar o valor do antecessor de  $x$  no conjunto  $S$  ou pode lançar uma exceção caso  $x$  não tenha sucessor ou caso  $x$  não esteja no conjunto.
- EMPTY(): retorna TRUE se, e somente se, o conjunto for vazio.
- SIZE(): retorna o tamanho do conjunto.

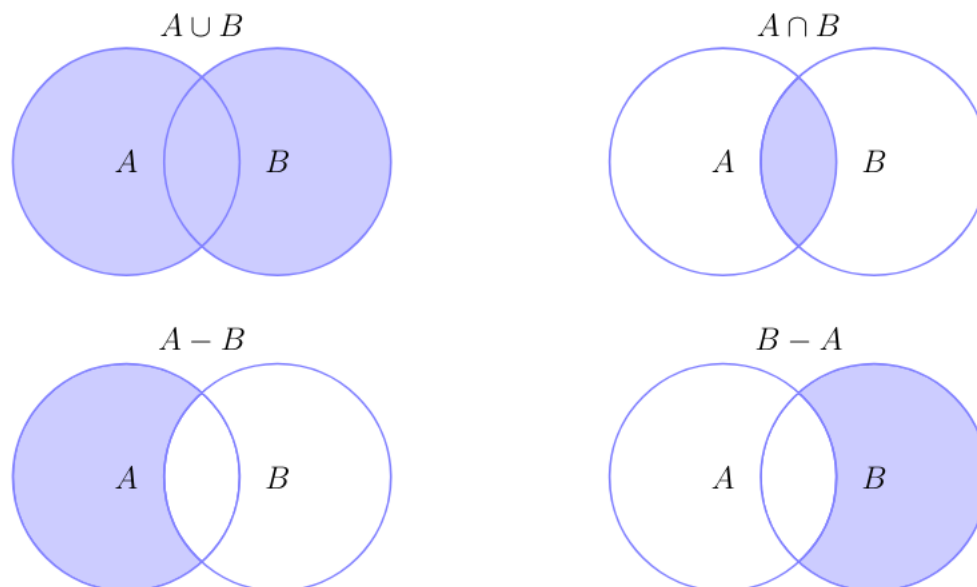
Nesta atividade, deverá ser desenvolvida uma aplicação que utilize uma estrutura de dados SET desenvolvida por você, usando como base uma árvore binária de busca AVL. É obrigatório o uso da árvore balanceada, pois as operações de inserção, remoção

e busca na estrutura de dados SET devem ter complexidade de tempo de pior caso  $O(\lg n)$ , onde  $n$  é a cardinalidade do conjunto.

Existem também operações binárias realizadas sobre dois conjuntos, que são muito realizadas por algoritmos, que são:

- **UNION**( $S, R$ ): dados conjuntos  $S$  e  $R$ , essa operação devolve o conjunto união  $S \cup R$ .
- **INTERSECTION**( $S, R$ ): dados conjuntos  $S$  e  $R$ , essa operação devolve o conjunto interseção  $S \cap R$ .
- **DIFFERENCE**( $S, R$ ): dados conjuntos  $S$  e  $R$ , essa operação devolve o conjunto diferença  $S - R$ . Lembre que  $S - R$  é o conjunto de elementos de  $S$  que não estão em  $R$ .

Estas três operações estão ilustradas nos diagramas de Venn da figura a seguir.



### Descrição dos requisitos

Implementar uma estrutura de dados conjunto dinâmico de inteiros (ou pode ser uma estrutura de dados genérica), satisfazendo as seguintes restrições:

- (1) O programa deve ser desenvolvido na linguagem de programação C++.
- (2) Deve usar o paradigma de programação orientado a objeto.
- (3) Todas as operações descritas acima devem ser fornecidas pelo seu programa.
- (4) As operações Insert, Erase, Contains, Minimum, Maximum, Successor e Predecessor devem TODAS ter complexidade de tempo  $O(\lg n)$  no pior caso.
- (5) Escreva um programa principal (main.cpp) com um menu de opções para que o usuário possa utilizar e testar todas as operações da estrutura de dados que você implementou. No seu programa, dê a possibilidade do usuário criar pelo menos três conjuntos para que ele possa testar as três operações binárias nesses conjuntos.