

Lista 6 - Conjuntos

ESTRUTURA DE DADOS I – Pedro Nuno Moura

Monitor: Victor Hugo Souza Wirz

Para as questões abaixo, considere a implementação de Conjunto Genérico baseado em Lista Encadeada vista em sala de aula.

ATENÇÃO: Para as questões abaixo, devem ser providas soluções contendo a melhor complexidade computacional possível. Além disso, deve ser informada e explicada a complexidade alcançada.

1) Crie métodos que retornem o menor e o maior elementos de um conjunto em tempo constante, isto é, cujas complexidades correspondam a $O(1)$.

2) Implemente um método que receba um conjunto `conj2` e verifique se o conjunto corrente, para o qual o método é chamado, é igual ao conjunto `conj2`. O método deve retornar `true` caso os dois conjuntos sejam iguais e `false`, caso contrário. O protótipo do método é o seguinte:

```
public boolean eIgual(ConjGenerico<T> conj2);
```

3) Desenvolva um método que receba um conjunto `universo` como parâmetro e calcule o complementar do conjunto corrente em relação a tal `universo`. Se o conjunto corrente possuir elementos de fora do conjunto `universo` informado, o método deve lançar uma exceção. O protótipo do método a ser implementado é o seguinte:

```
public ConjGenerico<T> complementar(ConjGenerico<T> universo);
```

4) Implemente um método que receba um conjunto `conj2` e retorne `true` se o conjunto corrente, para o qual método é chamado, é subconjunto de `conj2` e `false`, caso contrário. O método possui o protótipo abaixo:

```
public boolean eSubconjunto(ConjGenerico<T> conj2);
```

5) Implemente um método que receba um conjunto `conj2` e obtenha o conjunto diferença entre o conjunto corrente e `conj2`. O método deve retornar o conjunto resultante e deve seguir o protótipo abaixo:

```
public ConjGenerico<T> diferenca(ConjGenerico<T> conj2);
```

6) Implemente o método do Exercício 2 de maneira recursiva.

7) Dados dois conjuntos “A” e “B”, a Segunda Lei De Morgan diz que $(A \cup B)^c \rightarrow A^c \cap B^c$. Dessa maneira, pede-se que seja feito um método estático que, dados dois conjuntos `conj1` e `conj2`, aplique a Segunda Lei De Morgan a esses, retornando a interseção do complemento de `conj1` e do complemento de `conj2`. O método possui a seguinte assinatura:

```
public static ConjGenerico<T> aplicaDeMorgan(ConjGenerico<T> conj1,  
ConjGenerico<T> conj2);
```