NOTA: Esta ficha está dividida em duas partes (Parte I e Parte II), pretende-se que a primeira parte seja seguida com os slides da aula teórico-prática. Na segunda parte pretende-se que o aluno consiga realizar os exercícios pondo em prática a matéria abordada nos slides e praticada na Parte I. Devem ser consultados os slides: **`2019.ED.Aula10.pdf`** para a realização desta ficha prática.

Parte I

Exercício 1

Implementar uma `LinkedBinarySearchTree` como sugerido nos slides da aula teórica (atenção que os métodos `removeAllOccurrences` e `removeMin` ainda não estão implementados). Demonstre a utilização da `LinkedBinarySearchTree` através de um cenário à sua escolha.

Nota 1: o método `removeAllOccurrences` deverá remover todas as ocorrências de um determinado elemento passado por parâmetro.

Nota 2: o método `removeMin` deverá remover o menor elemento da árvore

Exercício 2

Implementar uma `ArrayBinarySearchTree` (em tudo semelhante à anterior, mas com recurso a um array em vez de uma lista ligada). Demonstre a utilização da `ArrayBinarySearchTree` através de um cenário à sua escolha (o mesmo definido no exercício 1).

Parte II

Exercício 1

O que é uma árvore binária de pesquisa?

Exercício 2

O que entende por árvore binária degenerada?

Exercício 3

Implementar uma lista ordenada com recurso a uma árvore binária de pesquisa.

Nota: Atenção que tem de preencher certos requisitos.

Exercício 4

Implementar uma árvore AVL (`AVLTree`).

Nota: Tenha em atenção o fator de balanceamento: "As árvores AVL acompanham a diferença de altura entre as sub-árvores direita e esquerda de cada nó".