

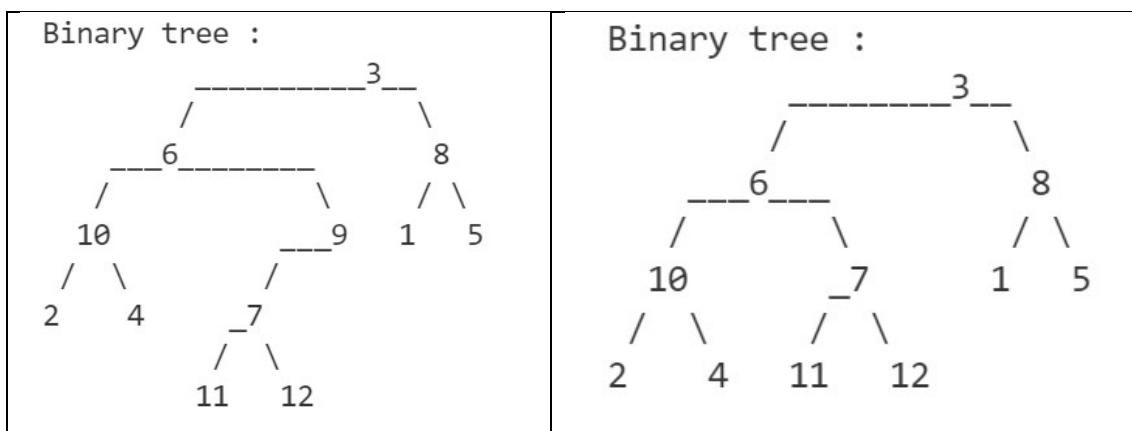
Prática 4 Estruturas de Dados 2

- Árvores
- Grafos

Exercício 4.1 - Árvores

Usando como base o código fornecido ([arvore_exemplo.py](#)), usando o módulo **binarytree**, crie uma árvore que replique a apresentada abaixo à esquerda.

Com base na árvore que criou, crie uma nova árvore eliminando o nó com o valor 9, mas mantendo o resto dos nós e fazendo com que a árvore tenha a representação da figura abaixo à direita. Seria mais fácil ou mais difícil eliminar o nó com o valor 8?



Exercício 4.2 - Previsão de palavras

Fazendo uso da(s) classe(s) em [Trie.py](#) desenvolva um programa que permita ao utilizador sempre que escreva as letras iniciais de uma palavra (do Português) ter informação de quais as palavras que são possíveis com esse início.

O programa deve continuar a processar palavras até o utilizador inserir um conjunto vazio de caracteres.

Deve começar por inserir numa instância da classe Trie um conjunto grande de palavras, sugerindo-se as listas de palavras em [palavras1.txt](#) (cerca de 8000 palavras) e [palavras2.txt](#) (mais de 300 mil palavras).

Exercício 4.3- Grafos

Instale o módulo **networkx** e, adaptando o exemplo que foi dado na TP, crie um grafo representando as distâncias entre pelo menos 10 cidades portuguesas, desenhe a sua representação gráfica e determine o caminho mais curto entre 5 pares de cidades à sua escolha. Verifique se os resultados estão de acordo com o esperado.

Sugere-se que a lista de cidades inclua “Aveiro”, “Porto”, “Braga”, “Viana do Castelo”, “Lisboa” e “Faro”.

Para as distâncias sugere-se a utilização de um serviço online, por exemplo <http://www.distanciasentrecidades.com/>.