

**Resolva cada exercício em folhas separadas**

**3.5 p<sup>tos</sup>**

1. Desenvolva um método que recebe um texto numa String e devolve um map com a média do número de caracteres das palavras que começam pela mesma letra. Por exemplo, para o texto:

“ O rato roeu a rolha da garrafa do rei”

o resultado deverá ser:

```
O 1.0  
r 4.0  
a 1.0  
d 2.0  
g 7.0
```

O método deverá ter a seguinte assinatura:

```
public Map<Character,Double> meanWords(String[] text)
```

**Nota:** Será valorizada uma resolução o mais eficiente possível e com o menor número de estruturas de dados

**3.5 p<sup>tos</sup>**

2. Considere o seguinte código:

```
public List<T> enigma(List<T> q) {  
  
    if (q.isEmpty())  
        return q;  
  
    T data = q.remove(0);  
    q = enigma(q);  
    q.add(data);  
    return q;  
}
```

- b) Explique o que faz o método acima apresentado. Considere a invocação do método enigma com o seguinte conteúdo: [2, 5, 6, 9]. Qual o resultado?
- c) Analise a função quanto à sua complexidade temporal. Justifique.

**5 p<sup>tos</sup>**

3. Adicione à classe TREE<E> um método que verifica se uma árvore BST é AVL.

**5 p<sup>tos</sup>**

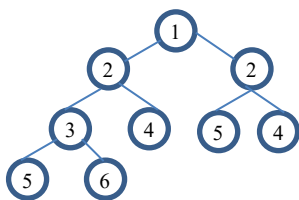
4. Um grafo conexo é um grafo em que todos os seus vértices são alcançáveis a partir de um qualquer outro vértice do grafo. Usando a representação **map de adjacência** implemente um método que, dado um **grafo não conexo**, devolve para cada vértice do grafo a respetiva lista de vértices não alcançáveis. O método a desenvolver deve obedecer à interface:

```
public static<V,E> Map<V,List<V>> vertsnotReachable(Graph<V,E> g)
```

**Resolva cada exercício em folhas separadas**

**3 p<sup>tos</sup>**

5. Implemente na classe `HeapPriorityQueue<K,V>` um método que devolve o índice da primeira ocorrência de um elemento na heap. Para a heap abaixo representada observe os dois exemplos:



Exemplos:

[1,2,2,3,4,5,4,5,6]

Value=5 → idx=5

value=6 → idx=8

value=9 → idx=-1

Considere a seguinte assinatura:

```
public int getIndElem(V value)
```