isep Instituto Superior de Engenharia do Porto

Estruturas de Informação
Exame Época Recurso

6 de Fevereiro 2019

Departamento de Engenharia Informática

Duração 2 horas

Resolva cada exercício em folhas separadas

3.5 p^{tos}

1. Pretende-se empilhar vários objetos com pesos específicos em paletes com igual capacidade de peso. Para tal elabore um método que recebe uma lista com os pesos dos objetos e atribui estes às paletes de modo a não exceder a capacidade destas e minimizar o número de paletes necessárias. Por exemplo para paletes com capacidade máxima 10 e a seguinte lista de pesos: [4, 8, 2, 1, 7, 6, 1, 4, 5, 2] uma possível alocação dos pesos poderá ser:

Paletes	Pesos
1	[4.0, 2.0, 1.0, 1.0, 2.0]
2	[8.0]
3	[7.0]
4	[6.0, 4.0]
5	[5.0]

O método deve devolver os vários pesos alocados a cada palete bem como a taxa média de ocupação total das paletes.

Considere a seguinte assinatura:

```
Double packing(Double Capac, List<Double> pesos, Map<Integer, LinkedList<Double>> paletes)
```

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

3.5 p^{tos}

2. Considere o seguinte código:

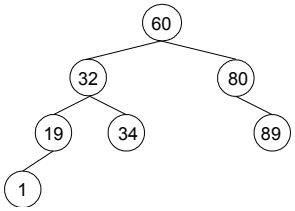
```
public static int mistery (String tt, String pp){

    for (int i = 0; i <= tt.length()-pp.length(); i++){
        int j = 0;
        while (j < pp.length() && tt.charAt(i + j) == pp.charAt(j)){
            j++;
        }
        if (j == pp.length())
            return i;
    }
    return -1;
}
```

a) Explique o que faz o método mistery acima apresentado.

b) Analise o método quanto à sua complexidade temporal. Seja preciso e justifique a resposta.

5 p^{tos}

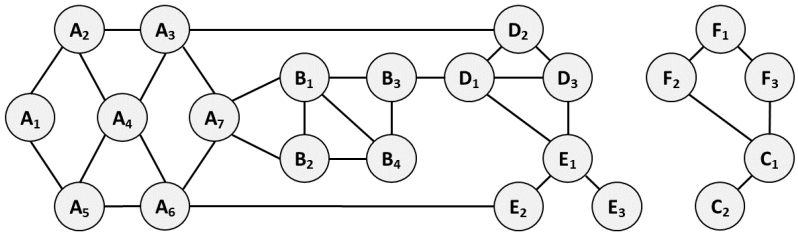
3. Adicione à classe TREE<E> um método faça uma **visita por nível inversa**, ou seja do último nível para o primeiro nível. Não é permitido usar qualquer método de visita da classe BST. Para a árvore a seguir apresentada o método deve devolver uma lista com os elementos: [1,19,34,89,32,80,60]



Resolva cada exercício em folhas separadas

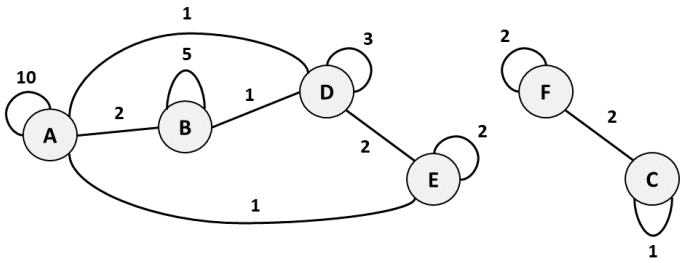
5 p^tos

4. Num grafo de uma rede social, aplicando um algoritmo de detecção de comunidades é possível agrupar os vértices do grafo em **grupos** de acordo com um determinado critério. Este algoritmo atribui a cada um dos vértices do grafo uma etiqueta para indicar vértices pertencentes ao mesmo grupo, por ex. Os vértices A1..A7 pertencem ao grupo A, B1..B4 ao grupo B e assim sucessivamente.



Vértice	Grupo
A_1	A
...	
A_7	A
B_1	B
...	
F_1	F
...	
F_3	F

Usando a representação **map de adjacência** implemente um método que, dado um grafo de uma rede social e o respetivo map de etiquetas de vértice/grupo, devolva outro grafo que constitui o seu **grafo sumário**. Os vértices do grafo sumário representam os grupos e o peso do ramo entre dois vértices (dois grupos) é dado pelo número de ligações entre os vértices de ambos os grupos, ex.: $peso_{A,D} = 1$ (ligação A_3-D_2), $peso_{A,B} = 2$ (ligações A_7-B_1, A_7-B_2),e $peso_{A,E} = 1$ (ligação A_6-E_2). Para além destes ramos o grafo sumário apresenta também lacetes cujo peso é metade das ligações entre vértices que pertencem ao mesmo grupo.

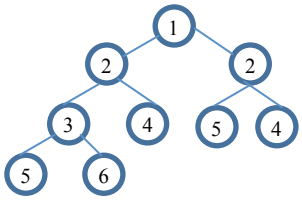


O método a desenvolver deve obedecer à interface:

```
public static<V,E> Graph<V,E> GraphSummary(Graph<V,E> g, Map<V,V> groupverts)
```

3 p^tos

5. Implemente um método que para uma qualquer heap retorna o número de elementos do último nível. Por exemplo para a heap [1,2,2,3,4,5,4,5,6] abaixo representada deve devolver 2.



Considere a seguinte assinatura:

```
public static<K,V extends Comparable<V>> int NumbElemsLastLevel(HeapPriorityQueue<K,V> heap)
```