isep Instituto Superior de Engenharia do Porto

Estruturas de Informação
Época Especial

4 de Setembro 2018

Departamento de Engenharia Informática

Duração 2 horas

Resolva cada exercício em folhas separadas

3.5 p^{tos}

1. Elabore um método que recebe duas listas com as coordenadas (x,y) de pontos no plano e retorna a maior distância euclidiana verificada entre os vários pontos. Considere que as listas possuem o mesmo tamanho.

```
public double EucliDistPoints (List<Integer> abcissas, List<Integer> ordenadas)
```

3.5 p^{tos}

2. Considere o seguinte método.

```
int binarySum (int[] data, int low, int high) {
    if (low > high)
       return 0;
    else if (low == high)
       return dat[low];
    else {
       int mid = (low + high) / 2;
       return binarySum(data, low, mid) + binarySum(data, mid+1, high);
   }
}
```

Analise o método quanto à sua complexidade temporal. Justifique.

5 p^{tos}

3. Considere um sistema de rega disposto em forma de árvore binária em que a quantidade de água disponibilizada no ponto inicial vai repartir-se por todos os canais de tal modo que, quando há uma bifurcação, a quantidade de água é dividida de igual forma por cada um dos canais. Adicione à classe TREE<E> um método que apresente a quantidade de água que chega a cada um dos nós folha.

```
private ArrayList<Double> waterInLeaves (Node<E> node, Double litrosIniciais)
```

5 p^{tos}

4. Considere um grafo que representa a planta de celas de uma prisão, das quais algumas celas contêm detector de movimentos e apenas algumas celas permitem o acesso ao exterior. Usando a representação map de adjacências pretende-se que elaborem um método de **plano de fuga** que para uma determinada cela origem encontre o caminho mais curto para a cela com acesso ao exterior mais próxima, sem passar por celas com detectores de movimentos. Admita que em cada cela é possível escavar um túnel para qualquer outra cela adjacente e note que se a cela origem ou as celas com saída para o exterior tiverem detectores de movimento o plano de fuga não será viável.

```
public LinkedList<Integer> escapePlan(Integer origCell)
```

3 p^{tos}

5. Implemente na classe HeapPriorityQueue<K,V> um método que altera a prioridade de todos os elementos de uma heap com uma determinada prioridade. O método deve retornar a heap resultante. Considere a seguinte assinatura:

```
public static <K,V extends Comparable<V>>  HeapPriorityQueue<K,V>
                      changePriorityElem(HeapPriorityQueue<K,V> heap, K key, K finalkey)
```