isep Instituto Superior de Engenharia do Porto

Estruturas de Informação

Época Normal

30 de Janeiro 2017

Departamento de Engenharia Informática

Duração 2 horas

Resolva cada exercício em folhas separadas

4 p^tos

1. Dada uma lista de N inteiros não repetidos e k centros, inteiros dessa lista, pretende-se criar k sublistas (k >= 2), de tal modo que em cada sublista ficam os inteiros mais próximos do respectivo centro, ou seja, os inteiros da lista inicial com a menor diferença absoluta relativamente ao centro.
   Como exemplo considere a seguinte lista L = { 2, 9, 7, 5, 10, 15, 6, 12, 3} e os centros C_1=3, C_2=6 e C_3=10. A lista original L será dividida nas três sublistas: L1 = {3, 2}, L2 = {6, 7, 5} e L3 = {10, 9, 15, 12} que devem ser devolvidas num contentor map que guarda os pares <centro, sublista>.

```
Map<Integer, LinkedList<Integer>> KsubLists(LinkedList<Integer> list,
                                            ArrayList<Integer> centers)
```

3 p^tos

2. Considere o seguinte método implementado na classe HeapPriorityQueue<K,V>:

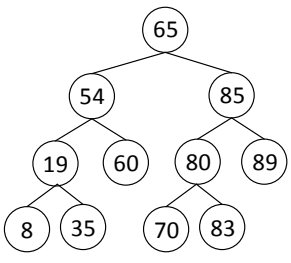
```
public HeapPriorityQueue<K,V> misterio () {

    HeapPriorityQueue<K,V> aux = new HeapPriorityQueue<>();
    Entry<K,V> e;

    int startIndex = heap.size() - 1;
    for (int j=startIndex; j >= 0; j--) {
        e = heap.get(j);
        K k = e.getKey();
        V v = e.getValue();
        aux.insert(k, v);
    }
    return aux;
}
```

   a) Explique o que faz o método acima apresentado.

   b) Analise o método quanto à sua complexidade temporal. Justifique.

4.5 p^tos

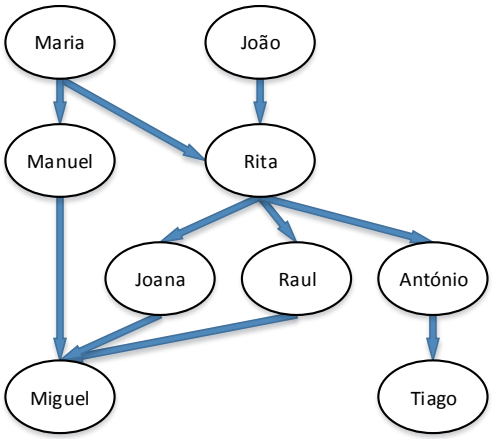
3. Adicione à classe TREE<E> um método que faz uma travessia numa árvore binária de acordo com a orientação de uma string de 0s e 1s, de tal modo que um zero desce na subárvore esquerda e um 1 desce na subárvore direita. O método deverá receber uma árvore binária e uma string de 0s e1s e devolver o elemento do nó da árvore correspondente ao último elemento da string, ou null no caso da string não terminar num nó válido. Por exemplo, para a árvore abaixo e com a string 101 o método devolve o elemento 83, com a string 110 devolve null.



Resolva cada exercício em folhas separadas

4.5 p^{tos}

4. Considere um grafo orientado e acíclico, onde cada vértice representa um funcionário e cada ramo x -> y representa semanticamente que o funcionário y somente pode ser promovido se o funcionário x também for. Por exemplo, no grafo da figura se pretendermos encontrar 2 promoções, há 2 listas possíveis: { Maria, João } ou { Maria, Manuel }. No caso de se pretenderem 3 promoções, as opções são { Maria, João, Manuel } ou { Maria, João, Rita }.



Usando a representação **map de adjacência** implemente um método que, dado um grafo e um número **n** de promoções, devolva uma lista com os funcionários a promover. O objetivo é somente encontrar uma das soluções. O método a desenvolver deve obedecer à interface:

```
List<String> calculaPromocoes(Graph<String,Integer> g, Integer n)
```

4 p^{tos}

5. Implemente na classe HeapPriorityQueue<K,V> uma função que coloca num vector a sub-heap correspondente a um índice (idx) de uma heap e devolve o número de elementos dessa sub-heap. O seguinte exemplo ilustra o pretendido:

```
public int getSubHeap(int idx, V[] vet)
```