isep Instituto Superior de Engenharia do Porto

Estruturas de Informação

Época Recurso

9 de Fevereiro 2017

Departamento de Engenharia Informática

Duração 2 horas

Resolva cada exercício em folhas separadas

4 p^{tos}

1. Pretende-se que implementem o jogo da roleta russa através de uma lista circular com N inteiros. Para tal, em cada iteração é gerado um número aleatório que corresponde à posição a ser removida na lista. O jogo continua na **posição seguinte à removida, no sentido dos ponteiros do relógio**, até restar apenas um inteiro na lista. Como exemplo considere a seguinte lista L = { 2, 9, 7, 5, 10, 15, 6, 12, 3 } e os números aleatórios:
   - Nº aleatório$_1$ = 3 (posição=3): 3 -> L_1 = { 2, 9, 7, 10, 15, 6, 12, 3 }
   - Nº aleatório$_2$ = 6 (posição=1): 6 -> L_2 = { 2, 7, 10, 15, 6, 12, 3 }
   - ....
   - Nº aleatório$_8$ = 1 (posição=8): L_8 = { 6 }

   Todas as sublistas deverão ser devolvidas num contentor map que guarda os pares posição na lista inicial removida e sublista <posição, sublista>. O método deve obedecer à seguinte assinatura:

```
Map<Integer, LinkedList<Integer>> roletaRussa(LinkedList<Integer> list)
```

   Considere as seguintes funções para gerar um número aleatório entre 0 e o tamanho da lista:

```
Random generator = new Random();
ind = generator.nextInt(list.size());
```

3 p^{tos}

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

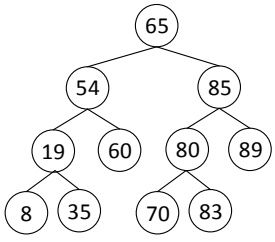
```
public Entry<K,V> mistério() {

      if (heap.isEmpty()) return null;
      Entry<K,V> answer = heap.get(0);
      swap(0, heap.size()-1);
      heap.remove(heap.size()-1);
      percolateDown(0);
      return answer;
}
```

   a) Explique o que faz o método acima apresentado. Considere a heap com o seguinte conteúdo: [2, 5, 6, 9, 7, 9, 20, 10, 12, 24, 8, 15, 13]. Qual o resultado (answer e heap)?

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

4.5 p^{tos}

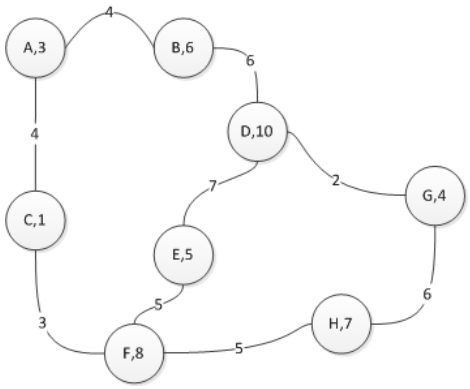
3. Adicione à classe TREE<E> um método que coloque numa lista o caminho completo desde a raiz até um determinado nó da árvore e devolva a profundidade desse caminho. A profundidade do caminho é a soma das profundidades de todos os nós que compõem o caminho. A profundidade de um nó p é o número de antecessores de p. Por exemplo, para a árvore abaixo o caminho da raiz até ao nó 83 será [65,85,80,83] com profundidade 6.



Resolva cada exercício em folhas separadas

4.5 p^{tos}

4. Considere uma rede de processamento que guarda informação sobre a capacidade de processamento disponível em cada nó bem como a capacidade de fluxo de informação entre os vários nós, representada através de pesos nas ligações entre os nós. O grafo bidirecional seguinte representa uma versão simplificada deste cenário:



Nesta rede, o nó "A" tem uma capacidade de processamento disponível de 3 e a ligação entre "A" e "B" tem uma capacidade de fluxo 4.

Escreva um método que receba dois nós e devolva numa lista a solução ou uma das soluções onde haja menos ligações entre esses dois nós. Pretende-se ainda que seja indicada a média de fluxo existente na respectiva ligação. Por exemplo entre os nós "A" e "G" a menor ligação é A->B->D->G com a correspondente média de fluxo 4.0.

```
double smaller_connection (No no1, No no2, LinkedList<String> lstNos)
```

4 p^{tos}

5. Escreva um método eficiente que verifica se dois heaps são equivalentes. Dois heaps são equivalentes se contêm as mesmas chaves e estas obedecem ao critério de prioridade do heap. Por exemplo, os dois heaps abaixo são equivalentes.