isep Instituto Superior de Engenharia do Porto

Estruturas de Informação

Época de Recurso

11 de Fevereiro 2015

Departamento de Engenharia Informática

Duração 2 horas

Resolva cada exercício em folhas separadas

6 pontos

1. Considere as seguintes classes para gestão dos seguros e apólices dos clientes de uma seguradora.

```
class Apolice
{
	private:
		int codCli;
		string nomeCli;
		list<string> lcc;    //coberturas complementares
		double premio;

	public:
		Apolice();
		Apolice(const Apolice& ap) ;
		…
};

class tipoSeguro
{
	private:
		int codSeg;
		list<string> lcb; //coberturas base

	public:
		…
};

class Seguradora : public map<tipoSeguro, list<Apolice> >
{
	private:
		string nome; // nome da empresa;
	public:
		…
};
```

Acrescente às classes **TODOS** os métodos necessários para realizar as seguintes operações:

a) Calcular o valor total de prémios a pagar por um dado cliente.

b) Inserir uma nova cobertura complementar para todos os clientes com um determinado tipo de seguro.

c) Acrescentar 15% ao prémio de todos os clientes com mais de três coberturas complementares

4 pontos

2. Considere os seguintes métodos:

```
bool outroMisterio (vector<int> v, int e){
  int i = 0;
  while (i < v.size() && v[i] != e) {
     i++;
  }
  if (i < v.size()) return true; else return false;
}
```

Resolva cada exercício em folhas separadas

```
void misterio (queue<int> qi, vector<int> vi, vector<int>& vo){

        while (!qi.empty()) {
                int e = qi.front();
                qi.pop();
                if (outroMisterio(vi, e))
                    vo.push_back(e);
        }
}
```

a) Diga qual o objetivo do método `outroMisterio`.

b) Faça a análise a complexidade temporal do método misterio. Justifique.

6 pontos

3. Uma distribuidora de jornais e revistas faz entregas a um conjunto de N pontos de venda, caracterizados pela morada e quantidade de exemplares a entregar. As rotas de distribuição têm associada a distância entre os pontos de venda.

a) Defina as classes necessárias à representação das ligações entre os diferentes pontos de venda recorrendo à classe *graphStlPath*.

b) Elabore um método que dado um ponto de venda devolva numa *stack* uma rota de distribuição com origem e destino nesse ponto, e que passe por todos os pontos de venda.

c) Elabore um método que devolva numa *queue* o(s) ponto(s) de venda cuja média de distância aos pontos de venda adjacentes é menor.

3 pontos

4. Defina na classe template tree<TN> o método LevelComplete, que verifica se um determinado nível da árvore está completo.

1 ponto

5. Considere a estrutura de informação designada por fila de prioridade ou HEAP organizado por máximos.

a) Apresente graficamente a HEAP correspondente à introdução dos elementos: 12, 7, 5, 20, 30, 10. Deve redesenhar a HEAP a cada introdução de um elemento.

b) Apresente graficamente a HEAP correspondente à eliminação do elemento da raiz da HEAP. Deve redesenhar a HEAP a cada alteração da HEAP após a eliminação.