isep Instituto Superior de Engenharia do Porto
Departamento de Engenharia Informática

Estruturas de Informação

Época Normal
29 de Janeiro 2015
Duração 2 horas

Resolva cada exercício em folhas separadas
As questões 1c) e 3c) são apenas para os alunos que mantiveram nota de frequência 2013/14

1. Considere a classe **Mensagem** que representa uma mensagem de e-mail e a classe **Servidor** que possui duas caixas de e-mail, uma para o envio e outra para a receção de mensagens dos seus utilizadores.

```
class Mensagem {
   private:
        string remetente;
        list<string> lista_destinatarios;
        string assunto;
        string texto;

    public:
        Mensagem();
        Mensagem(const Mensagem& m);
        ~Mensagem();
        …
};

class Servidor {
   private:
         vector<Mensagem> caixaIn;  // INBOX
         vector<Mensagem> caixaOut; // OUTBOX

   public:
        Servidor();
        Servidor(vector<Mensagem>& cxin, vector<Mensagem>& cxot);
        ~Servidor();
        …
};
```

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

a) Colocar numa lista todas as mensagens enviadas por um dado utilizador, ordenadas por ordem alfabética do assunto.
b) Devolver num vetor a(s) mensagem(s) enviadas com o maior nº de destinatários.
c) Apagar da caixa de entrada as mensagens com um determinado assunto.

2. Considere os seguintes métodos:

```
int outroMisterio (vector<int> v, int x, int y){
   int r = 0;

   for (int i = x; i < x + y; i++){
      r = r + v[i];
   }
   return r / y;
```

```
void misterio (vector<int> vi, vector<int>& vo){
   int s = 3;

   for (int i = 0; i<vi.size() - s; i++){
      vo[i] = outroMisterio(vi, i, s);
   }
}
```

a) Diga qual o objetivo do método *outroMisterio*.
b) Faça a análise a complexidade temporal do método *misterio*. Justifique.

Resolva cada exercício em folhas separadas
As questões 1c) e 3c) são apenas para os alunos que mantiveram nota de frequência 2013/14

3. Considere a seguinte tabela com exemplos de informação de custo e respetiva duração em minutos para ligações aéreas entre vários aeroportos:

Viagem			Custo	Duração
Porto	→	Lisboa	17,00 €	60
Porto	→	Faro	22,00 €	60
Porto	→	Madrid	26,00 €	70
Porto	→	Barcelona	23,00 €	100
Porto	→	Paris	148,00 €	135
Porto	→	Londres	45,00 €	145
Porto	→	Moscovo	850,00 €	320
Lisboa	→	Barcelona	98,00 €	105
Lisboa	→	Madrid	29,00 €	75
Lisboa	→	Porto	17,00 €	55

Viagem			Custo	Duração
Lisboa	→	Faro	151,00 €	40
Lisboa	→	Paris	127,00 €	150
Lisboa	→	Londres	35,00 €	160
Lisboa	→	Moscovo	733,00 €	355
Faro	→	Porto	37,00 €	70
Faro	→	Barcelona	98,00 €	75
Faro	→	Paris	138,00 €	180
Faro	→	Londres	296,00 €	160
Madrid	→	Barcelona	23,00 €	75
Madrid	→	Paris	61,00 €	125

Viagem			Custo	Duração
Barcelona	→	Paris	56,00 €	120
Barcelona	→	Roma	40,00 €	115
Roma	→	Paris	56,00 €	120
Roma	→	Londres	63,00 €	165
Paris	→	Londres	50,00 €	70
Paris	→	Roma	120,00 €	120
Londres	→	Moscovo	221,00 €	220
Paris	→	Moscovo	119,00 €	260
Roma	→	Moscovo	201,00 €	280
Madrid	→	Moscovo	523,00 €	310

Através da combinação de várias ligações, podem obter-se itinerários com vários preços e tempos de viagem entre duas cidades.

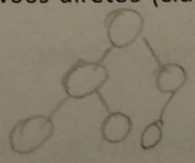
a) Utilizando a classe *graphStlPath* defina as classes necessárias para guardar a informação exemplificada na tabela através de um único grafo dirigido. Não é necessário indicar os métodos das classes.

b) Elabore um método que dada uma cidade de origem devolva numa *queue* todos os itinerários que envolvam mais do que um voo, sem repetição de cidades, cujo preço final e duração total de voo não excedam valores pré-definidos.

Alguns exemplos de itinerários válidos para partidas do Porto até 240,00€ e 250 minutos de duração:

>Porto>Faro>Paris (160€, 240 min.)
>Porto>Lisboa>Madrid>Barcelona (69€, 210 min.)
>Porto>Faro>Barcelona>Roma (160€, 250 min.)

c) Faça um método que para uma cidade origem indique todos os voos diretos (cidade destino, custo, duração).



4. Acrescente à classe tree<TN> um método que devolve a quantidade de nós completos cujo valor corresponde à média dos seus nós filhos.

5. Considere a estrutura de informação designada por fila de prioridade ou HEAP. Ilustre a execução do algoritmo heapsort sobre os valores 1, 4, 10, 2, 23, 5, 20, 22. Considere que o heap está organizado por máximos.