



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**  
**AD2 de Programação OO**  
**2º semestre de 2017**

**EXERCÍCIO (ENTREGAR OS ARQUIVOS EM MÍDIA, PARA FINS DE TESTE, JUNTAMENTE COM A AD IMPRESSA):**

Considere que sua empresa de software seja contratada pelos Correios para encontrar padrões para as soluções já implementadas para o problema de entrega de correspondências. Este problema consiste na procura de um caminho que, começa numa cidade de origem (isto é, a cidade onde as correspondências estão armazenadas) já definida, dentre várias, visita cada cidade **SOMENTE** uma vez e regressa à cidade inicial.

Seu programa deve receber, como parâmetro de entrada, um arquivo contendo as soluções já implementadas pelos Correios, e deve retornar, no arquivo **resp-<nome do arquivo de entrada>**, os melhores trechos entre duas cidades que já foram usados mais de uma vez (em ordem decrescente, mostrando sempre os melhores trechos primeiro).

Um EXEMPLO de arquivo de entrada neste formato seria:

**RJ/SP/VIX/BH/NIT/RJ**  
**RJ/VIX/SP/BH/NIT/RJ**  
**SP/BH/NIT/RJ/VIX/SP**  
**BH/NIT/RJ/VIX/SP/BH**

Para o exemplo supracitado, o arquivo de resposta seria composto dos seguintes trechos:

**NIT/RJ**           -> 4  
**BH/NIT**           -> 4  
**RJ/VIX**           -> 3  
**VIX/SP**           -> 3  
**SP/BH**           -> 3

Com essa resposta obtida pelo seu software, os Correios podem gerar novas soluções, como, por exemplo, **NIT/RJ/VIX/SP/BH/NIT**, que podem auxiliar na resolução desse problema. **PORÉM, A GERAÇÃO DESSAS NOVAS SOLUÇÕES NÃO DEVEM SER IMPLEMENTADAS NESTA AD2.**

**LEMBRE-SE: SEU PROGRAMA DEVE EXECUTAR COM QUAISQUER DADOS INFORMADOS COMO PARÂMETROS DE ENTRADA. SE O SEU PROGRAMA RESOLVER SOMENTE O PROBLEMA DO EXERCÍCIO SUPRACITADO, SUA QUESTÃO SERÁ TOTALMENTE DESCONTADA.**

RESPOSTA:

```
import java.io.*;

class no{
    String origem, destino;
    int cont;
    no prox;

    no(String o, String d){
        origem = o;
        destino = d;
        cont = 1;
        prox = null;
    }

    public String toString(){
        return origem + "/" + destino + "\t-> " + cont;
    }
}

class lista{
    no prim;

    lista(){ prim = null; }

    no busca(String o, String d){
        no p = prim;
        while(p != null){
            if ((o.compareToIgnoreCase(p.origem) == 0) &&
                (d.compareToIgnoreCase(p.destino) == 0)) break;
            p = p.prox;
        }
        return p;
    }

    void insere(String o, String d){
        no p = busca(o,d);
        if(p != null){
            p.cont++;
            return;
        }
        p = new no(o,d);
        p.prox = prim;
        prim = p;
    }

    public String toString(){
        String resp = "";
        no p = prim;
```

```

        while (p != null){
            resp += p.toString() + "\n";
            p = p.prox;
        }
        return resp;
    }
}

```

```

public class AD2_POO_2017_2{
    public static void main(String[] args) throws IOException {
        BufferedReader in = new BufferedReader(new FileReader(args[0]));
        BufferedWriter out = new BufferedWriter(new FileWriter("resp-" +
args[0]));
        String s;
        lista l = new lista();
        try{
            while((s = in.readLine()) != null){
                String vs[] = s.split("/");
                int i, n = vs.length;
                for(i = 0; i < (n - 1); i++) l.insere(vs[i], vs[i + 1]);
            }
            Ordena(l);
            RetiraTrechosTamanhoUnitario(l);
            out.write(l.toString());
        }catch (Exception e){
            System.out.println("Excecao leitura ou escrita...\n");
        }finally{
            in.close();
            out.close();
        }
    }
}

```

```

static void Ordena(lista l){
    if(l.prim != null){
        for(no p = l.prim; p != null; p = p.prox){
            no maior = p;
            for(no q = p.prox; q != null; q = q.prox)
                if (maior.cont < q.cont) maior = q;
            if(maior != p) troca(maior, p);
        }
    }
}

```

```

static void troca(no maior, no p){
    String aux = maior.origem; //troca origem
    maior.origem = p.origem;
    p.origem = aux;
    int temp = maior.cont; //troca cont
    maior.cont = p.cont;
    p.cont = temp;
}

```

```
    aux = maior.destino; //troca destino
    maior.destino = p.destino;
    p.destino = aux;
}

static void RetiraTrechosTamanhoUnitario(lista l){
    if(l.prim == null) return;
    if(l.prim.cont == 1){
        l.prim = null;
        return;
    }
    no p = l.prim;
    while(p.prox.cont != 1) p = p.prox;
    p.prox = null;
}
}
```