

 <p>UNIVERSIDADE DE COIMBRA FACULDADE DE CIÊNCIAS E TECNOLOGIA <i>Departamento de Engenharia Informática</i></p>	<p>Algoritmos e Estruturas de Dados Trabalho nº 3 2012-2013 – 2º Semestre</p> <p><b>Submissão Mooshak Tarefas A e B:</b> 18 de Abril de 2014 23:00.</p> <p><b>Submissão Mooshak Tarefas C:</b> 2 de Maio de 2014 23:00.</p> <p><b>Entrega Tarefa D:</b> 7 de Maio de 2014 17:00 no cacifo do docente das aulas teóricas.</p>
<p><i>Nota Importante: A fraude denota uma grave falta de ética e constitui um comportamento não admissível num estudante do ensino superior e futuro profissional licenciado. Qualquer tentativa de fraude pode levar a anulação da componente prática tanto do facilitador como do prevaricador.</i></p> <p><i>A utilização de código disponível na net, relativo a algoritmos de ordenamento, desde que não infrinja direitos de autor é permitido neste trabalho. É assumido que este código foi compreendido com nível de detalhe idêntico ao esperado se tivesse sido desenvolvido pelo aluno. A explicação do código é matéria de avaliação desta ficha.</i></p> <p><i>**Só são aceites para defesa trabalhos para os quais os alunos tenham entregue o relatório dentro dos prazos definidos.**</i></p>	

### Objectivos:

Estudo de algoritmos de ordenamento incluídos no plano da Disciplina.

### Tarefas

- Tarefa A – Ordenamento por Inserção
- Tarefa B – Ordenamento por chave à escolha do aluno
- Tarefa C – Ordenamento por base à escolha do aluno
- Tarefa D – Relatório

---

A escolha de um algoritmo de ordenamento representa um compromisso entre eficiência temporal, espaço de memória ocupada e simplicidade do algoritmo, não havendo um que seja universalmente superior a todos os outros.

Na escolha do algoritmo a adoptar é importante ter em atenção a utilização específica que este vai ter, nomeadamente o número de chaves a ordenar e o número de vezes que o programa vai ser utilizado.

---

A empresa NovasRotas Lda dedica-se ao desenvolvimento de software para gestão de frotas de veículos. O software utilizado envolve a existência de equipamento de localização a bordo das viaturas, das quais são recebidos registos (pontos recolhidos pelo receptor de GPS) com os seguintes campos:

- Identificador do registo (“integer”)
- Identificador da viatura (“integer”)
- “timestamp” (AAAA-MM-DD HH:MM:SS)
- latitude (“float”)
- longitude (“float”)

A seguir apresenta-se uma amostra dos registos recebidos, em que a primeira linha contém o nome das colunas (os ficheiros de entrada de teste não contém a linha de cabeçalho):

```
id,id_vehicle,date,latitude,longitude
1245619,879,2009-12-09 00:00:47,38.832,-9.09474
1245618,320,2009-12-09 00:00:53,38.7401,-9.22841
1245617,720,2009-12-09 00:00:55,38.7206,-9.12267
```

### O Problema

A empresa pretende identificar as zonas onde há maior passagem de veículos. Uma zona é definida por todas as latitudes/longitudes com o mesmo par de valores latitude/longitude truncados ao nível da centésima de grau.

Neste sentido deve desenvolver um programa que faça a listagem das zonas seguida do número de pontos encontrados nessa zona. A listagem deve compreender as zonas organizadas por ordem crescente, primeiro de latitude e depois de longitude.

### A Abordagem

Para este problema, a abordagem que se pretende passa por dois passos:

1. ordenamento de todos os registos por latitude seguida de longitude (ambas expressas em graus e truncadas ao nível da centésima);
2. varrimento da lista ordenada para contagem do número de pontos associados a cada par latitude/longitude.

Nota: as colunas com identificador do registo; identificador da viatura e “timestamp” (AAAA-MM-DD HH:MM:SS) não vão ser utilizadas.

Nas tarefas de ordenamento:

- Para as medições de tempo deve adoptar soluções que minimizem a contaminação do tempo de execução pelo tempo de leitura e escrita dos dados, medindo dentro do possível só os tempos de ordenamento. Se em alguma situação tal não for possível deve ser identificada.
- A formação da chave de ordenamento dos registos de entrada deve ter em conta o método de ordenamento utilizado.

### Tarefa A :: ordenamento por inserção

- Ordene os registos por latitude e longitude. De seguida faça a sua contagem por zona.
- Teste no mooshak.

### **Tarefa B :: ordenamento eficiente**

- Decida sobre o algoritmo de ordenamento baseado em chaves temporalmente mais eficiente. Substitua o algoritmo de ordenamento da tarefa A por este.
- Teste no mooshak.
- Teste no ambiente de desenvolvimento com a amostra que lhe vai ser disponibilizada.

### **Tarefa C :: ordenamento eficiente (o mesmo enunciado da tarefa B mas recorrendo a um algoritmo de ordenamento por base)**

### **Tarefa D :: relatório**

O relatório deve compreender quatro partes:

#### **1. Avaliação Experimental**

*Medições de desempenho na forma de gráfico para os algoritmos considerados nas tarefas A a C, utilizando a amostra 6 (a disponibilizar), a 10%, 20%, 40%, 60%, 80%, 100% dos dados desta amostra<sup>1</sup>.*

#### **2. Tomada de Decisão 1 (max. 1000 caracteres)**

*Justificação da decisão tomada na escolha do algoritmo de ordenamento por chave.*

#### **3. Tomada de Decisão 2 (max. 1000 caracteres)**

*Justificação da decisão tomada na escolha do algoritmo de ordenamento por base.*

#### **4. Tomada de Decisão 3 (max. 1000 caracteres)**

*Explique alternativas à abordagem definida para este problema e as suas implicações em termos de complexidade temporal e espacial da(s) solução(ões) alternativas propostas.*

## **Referências**

Introduction to Algorithms, 2nd Edition

Thomas H. Cormen, Charles Leiserson, Ronald Rivest, Clifford Stein

The MIT Press, 2001, 2005

Algorithms in C, Third Edition

Parts 1-4

Robert Sedgewick

Addison-Wesley

Materiais da disciplina de AED (2013-2014)

---

<sup>1</sup> Para o algoritmo de ordenamento por inserção, caso os tempos de execução se tornem demasiado longos, terminar as medições de tempo com uma porção da amostra que julgue razoável.

## Testes no Mooshak

### Input

O *input* é composto por uma linha com o número de pontos a considerar. Seguido das linhas com os registos dos pontos GPS produzidos pelos veículos.

**Deve ainda ter em atenção, nas opções tomadas, de que se pretende que estes algoritmos manipulem ficheiros com um número de registos da ordem de  $10^6$ , com quantidades consideráveis de chaves repetidas, com comprimento da chaves variável e em que se privilegia a rapidez de processamento relativamente à quantidade de memória ocupada.**

### Output

O output deverá compreender um conjunto de linhas, cada linha com a definição de cada zona e do número de pontos GPS detectados nessa zona ao longo de todo o ficheiro de *input*.

A definição de cada zona é feita em termos de latitude e longitude em graus e truncada na centésima de grau.

As zonas devem ser apresentadas por ordem crescente, primeiro de latitude e depois de longitude.

**Nota: tanto a última linha de input como de output terminam com o símbolo de mudança de linha.**

### Exemplo de *input*

```
8[\n]
1245619,879,2009-12-09 00:00:47,38.832,-9.09477[\n]
1245618,320,2009-12-09 00:00:53,38.7401,-9.22841[\n]
1245619,674,2009-12-09 03:00:47,38.836,-9.0956[\n]
1245617,720,2009-12-09 00:00:55,38.7206,-9.12267[\n]
1245619,674,2009-12-09 03:50:47,38.831,-9.0916[\n]
1245618,320,2009-12-09 00:00:59,38.7402,-9.22849[\n]
1245618,320,2009-12-09 00:00:53,38.7401,-9.12861[\n]
1245617,120,2009-12-09 10:00:55,38.7296,-9.12207[\n]
```

### Exemplo de *output*

```
38.72,-9.12,2[\n]
38.74,-9.22,2[\n]
38.74,-9.12,1[\n]
38.83,-9.09,3[\n]
```

Bom trabalho,

*Carlos Lisboa Bento*