

DEI/FCTUC – Universidade de Coimbra
Algoritmos e Estruturas de Dados, 2010/2011
Enunciado Trabalho Prático TP4 v1.0

Objectivos:

Estudo dos vários algoritmos de ordenamento incluídos no plano de estudos da Disciplina.

Exercícios

- Problema A – Por Troca
- Problema B – Merge Sort
- Problema C – Quick Sort

Prazo limite de submissão meta A no Mooshak:	26 de Abril de 2011, 12h00
Prazo limite de submissão meta B no Mooshak:	3 de Maio de 2011, 12h00
Prazo limite de submissão meta C no Mooshak:	17 de Maio de 2011, 12h00
Prazo limite de entrega do Relatório:	20 de Maio de 2008 18h00 (cacifo do docente aula TP)

Os alunos deverão consultar previamente o texto de apoio sobre o Mooshak disponível no sítio internet da disciplina.

MUITO IMPORTANTE:

- **O relatório do trabalho deve ser realizado no formulário que vai ser disponibilizado. Texto para além do espaço definido não será considerado. DEVE SER MANUSCRITO com excepção da secção de código.**
- **Só os alunos que entregam o relatório do trabalho no cacifo do respectivo docente até às 18:00 do dia 20 de Maio de 2010 são admitidos à avaliação deste trabalho.**
- **A falta à sessão de avaliação, de acordo com inscrição a realizar na secretaria do DEI, a menos de justificada por doença, comparência em acto oficial ou facto similar, implica também a não consideração deste trabalho para efeitos de avaliação.**
- **Os alunos, estudantes trabalhadores, podem optar por ter os trabalhos avaliados em conjunto, em data a definir oportunamente.**
- **A folha de inscrição para avaliação está disponível a partir do dia 16 de Maio na secretaria do DEI.**

Problemas A, B, C, D: *Algoritmos de Ordenamento*

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

Na escolha do algoritmo a adoptar é importante atender à utilização específica que este vai ter, nomeadamente a dimensão dos ficheiros a ordenar e o número de vezes que o programa vai ser utilizado e se necessitamos ou não de ter um ordenamento estável.

Neste trabalho é pedido aos alunos que implementem os algoritmos:

- Ordenamento por Troca (Problema A)
- Merge Sort (Problema B)
- QuickSort (Problema C)

e que analisem estes algoritmos segundo diversos parâmetros indicados no formulário do relatório.

Problema

Considere que está a desenvolver um serviço para uma rede social para partilha de Pontos de Interesse (POIs) sobre o espaço urbano (Fig. 1).

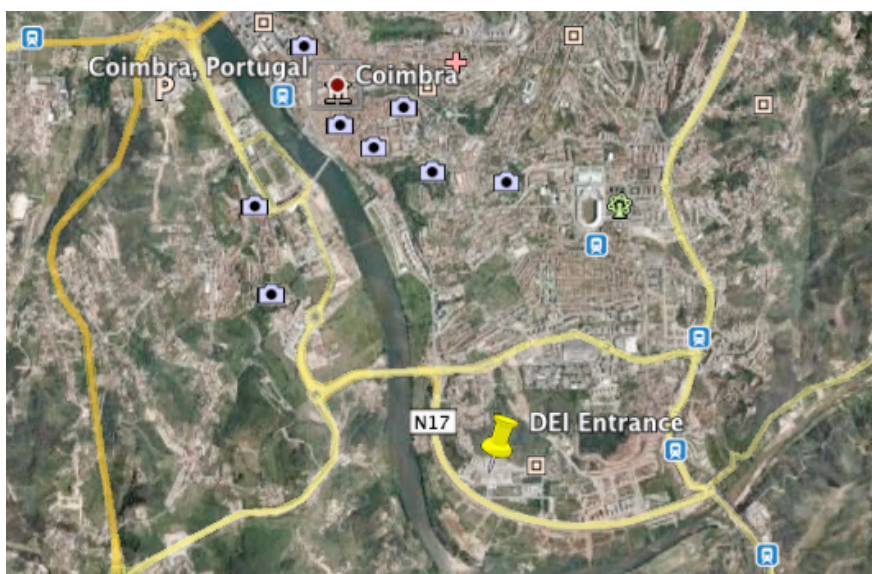


Fig. 1 – Pontos de interesse.

Os utilizadores da rede inserem os POIs no formato: $G_{lat} M_{lat} S_{lat} G_{lon} M_{lon} S_{lon} TEXTO$ com $G M S$, respectivamente, graus minutos e segundos para a latitude e longitude e $TEXTO$ uma sequência alfanumérica com o comprimento máximo de 40 caracteres. Nos segundos a parte decimal é truncada.

Uma das funcionalidades que o serviço deve ter é para aqueles POIs que têm a mesma localização eliminar todos menos o primeiro na sequência de inseridos.

Pretende-se recorrer ao ordenamento dos POIs com vista à eliminação eficiente dos repetidos.

É assumido que o ordenamento é feito primeiro por latitude e depois por longitude (esta ordem não é relevante para o problema em questão, mas deve ser considerada para efeito da submissão no Mooshak).

Input

O *input* é composto por uma linha com o número de elementos a ordenar. Seguido das linhas compostas por G_{lat} M_{lat} S_{lat} G_{lon} M_{lon} S_{lon} e uma cadeia de caracteres alfanuméricos com no máximo 40 posições.

Output

O output deverá compreender um conjunto de linhas, cada linha com a localização G_{lat} M_{lat} S_{lat} G_{lon} M_{lon} S_{lon} para a latitude e longitude e uma cadeia de caracteres.

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

Exemplo de *input*

```
4
40 12 31 8 25 36 Estação da Espadaneira
40 14 16 8 25 46 Estacao da Pedrulha
40 11 15 8 24 52 Polo II
40 14 16 8 25 46 Rui dos Leitoes
```

Exemplo de *output*

```
40 11 15 8 24 52 Polo II
40 12 31 8 25 36 Estação da Espadaneira
40 14 16 8 25 46 Estacao da Pedrulha
```

Exemplo de *input*

```
20
38 30 30 6 10 10 qasdfredcv
40 30 30 8 10 10 nmmnbvcxz1
40 30 30 7 10 10 kjnbhgtfed
40 30 30 8 10 10 zlkjhgfdsa
40 30 30 7 10 10 yuiopasdfg
41 30 30 6 10 10 ujmkiokjn
40 30 30 6 10 10 kjhgfdsapo
40 30 30 7 10 10 fghjklzxcv
40 30 30 8 10 10 uiopasdfgh
39 30 30 6 10 10 nhyujmkioi
40 30 30 8 10 10 trewqasdfr
38 30 30 6 10 10 fredcvfrtg
41 30 30 7 10 10 zxcvbnmmnb
41 30 30 7 10 10 qasdfredcv
39 30 30 7 10 10 tgbnhyujmk
40 30 30 8 10 10 redcvfrtgb
40 30 30 6 10 10 kjhgfdsapo
38 30 30 6 10 10 dfredcvfrt
40 30 30 6 10 10 iopasdfghj
40 30 30 8 10 10 ytrewqasdf
```

Exemplo de *output*

38	30	30	6	10	10	qasdfredcv
39	30	30	6	10	10	nhyujmkioi
39	30	30	7	10	10	tgbnhyujmk
40	30	30	6	10	10	kjhgfdsapo
40	30	30	7	10	10	kjnbhgtfed
40	30	30	8	10	10	nmnbnvcxzl
41	30	30	6	10	10	ujmkioikjn
41	30	30	7	10	10	zxcvbnmmnb

Relatório TP4 AED 2010/2011

Nome:

Nome no Mooshak:

Nº Estudante:

email:

TP (inscrição):

Nº de horas de trabalho (TP+PL+FSA):

(um dia aprox. 8 h de trabalho)

(A Preencher pelo Docente) Relatório: [0] [1] [] [5] [] [10]

(A Preencher pelo Docente) Código: [0] [1] [] [5] [] [10]

(A Preencher pelo Docente) Defesa Indiv.: [0] [2] [4] [6] [8] [10]

(A Preencher pelo Docente) CLASSIF. : [] []

Estudo Experimental de Complexidade Temporal (em ns)

(sequência de input aleatória)

Algoritmo	Nº de Entradas			
	50000	100000	150000	200000
Troca				
Merge Sort				
QS				

(sequência de input por ordem crescente)

Algoritmo	Nº de Entradas			
	50000	100000	150000	200000
Troca				
Merge Sort				
QS				

(sequência de input por ordem decrescente)

Algoritmo	Nº de Entradas			
	50000	100000	150000	200000
Troca				
Merge Sort				
QS				

(sequência de input com todas as chaves iguais)

Algoritmo	Nº de Entradas			
	50000	100000	150000	200000
Troca				
Merge Sort				
QS				

TRÊS aspectos a salientar em cada um dos algoritmos:

Troca

Merge Sort

Quick Sort

Código: