



## Laboratório 4 - Ordenação

Semana de 6 a 10 de novembro de 2017

Duração: 2 horas

### 1ª Parte

Os ficheiros `main4a.c` e `word.c` contêm uma solução quase completa para o problema da contabilização do número de palavras distintas num ficheiro de texto. Este problema foi originalmente apresentado no Laboratório 2. Na versão apresentada em `main4a.c` é utilizada uma estrutura de dados baseada em tabelas, sendo que cada elemento da tabela aponta para uma estrutura, a qual contém um inteiro e um ponteiro para aceder a uma cadeia de caracteres (tipo `t_palavra`). Pretende-se neste laboratório não só a contabilização do número de palavras diferentes mas também a sua ordenação, que poderá ser realizada utilizando um dos seguintes critérios:

1. ordem alfabética das palavras
2. número de ocorrências
3. comprimento das palavras

O critério de ordenação é especificado pelo utilizador na linha de comando. Para ordenação por ordem alfabética deverá ser usado o comando

```
lab04a ficheiro.txt -alfabetica {-a|-d},
```

onde os parâmetros `a` e `d` indicam, respectivamente, ordenação ascendente e ordenação descendente. Para ordenação por número de ocorrências deverá ser usado o comando

```
lab04a ficheiro.txt -ocorrencias {-a|-d}
```

e para ordenação pelo comprimento de palavras deverá ser usado o comando

```
lab04a ficheiro.txt -comprimento {-a|-d}.
```

Por omissão, a ordenação deverá ser alfabética e ascendente. Deve usar o algoritmo de ordenação implementado na função `sort()`.

Para que a solução apresentada fique completa falta apenas: i) completar as funções de comparação; ii) chamar a função de ordenação `sort()` no programa principal; e iii) libertar a memória alocada. Por fim, pretende-se ainda o estudo experimental da complexidade temporal do algoritmo de ordenação.

1. Veja cuidadosamente o código que lhe é fornecido, até compreender como está organizado, qual o algoritmo subjacente e de que forma os dados são guardados.
2. Identifique o algoritmo de ordenação implementado pela função `sort()`.
3. Verifique os cabeçalhos das funções de interface presentes nos ficheiros `word.h` e `sort.h`. Repare que o último argumento da função `sort()` é uma função.
4. Use o comando `make` e o ficheiro `Makefile` (fazendo as alterações necessárias) para compilar o programa.
5. Escreva as restantes funções de comparação para implementar todos os critérios de ordenação.
6. Teste o programa com os diferentes critérios de ordenação e diferentes ficheiros de entrada.
7. Altere o programa de modo a avaliar o número de instruções efectuadas. Usando os ficheiros de texto disponíveis na sua área, mostre que o algoritmo é  $\mathcal{O}(n^2)$ .

## 2ª Parte

Na 2ª parte deste laboratório pretende-se fazer a inserção ordenada de registos numa estrutura de dados do tipo lista simples. Cada registo é composto por 4 campos: **id**, **age**, **height** e **weight**, todos do tipo **int**. Nos ficheiros **LinkedList.h** e **LinkedList.c** encontram-se definidos os protótipos para manipulação de uma lista simples de **Items** e as suas implementações, respectivamente, sendo o tipo **Item** definido no ficheiro **defs.h**. Nos ficheiros **Entry.h** e **Entry.c** encontram-se definidos os protótipos e a respectiva implementação das definições e funções para manipulação de registos. No ficheiro **lab04b.c** encontram-se as funções necessárias para ler registos de um ficheiro de dados, criar a lista simples correspondente (na ordem inversa), e escrever no ficheiro de saída essa mesma lista.

Modifique os ficheiros que entender necessários de forma a que a inserção na lista se faça de forma ordenada, podendo a ordenação ser realizada por qualquer um dos campos, por ordem ascendente, ou descendente. O critério de ordenação é especificado pelo utilizador na linha de comando. Para ordenação usando o campo **id** deverá ser usado o comando

```
lab04b ficheiro.txt -id {-a|-d},
```

onde os parâmetros **a** e **d** indicam, respectivamente, ordenação ascendente e ordenação descendente. Para ordenação pelo campo **age** deverá ser usado o comando

```
lab04b ficheiro.txt -age {-a|-d}.
```

Para ordenação pelo campo **height** deverá ser usado o comando

```
lab04b ficheiro.txt -height {-a|-d}
```

e para ordenação pelo campo **weight** deverá ser usado o comando

```
lab04b ficheiro.txt -weight {-a|-d}.
```

Por omissão, considere ordenação por **id** ascendente.