

Proposta de Trabalho / Project Proposal : Recenseamento / Census

Algoritmos e Estruturas Dados I / Algorithms and Data Structures I

José Manuel Torres

jtorres@ufp.edu.pt

Outubro de 2016 / October 2016

Universidade Fernando Pessoa

Faculdade de Ciência e Tecnologia / Faculty of science and technology

1 Descrição do Problema

A tarefa de programação proposta tem como finalidade implementar, parcialmente, alguns algoritmos de processamento de um sistema de informação de recenseamento de pessoas de uma freguesia de Portugal. As pessoas são descritas pelo seguinte conjunto de propriedades. Note-se que as propriedades são todas do tipo string.

```
/**
 * @brief Struct describing the several fields about each person
 * Detailed description for struct pessoa. All fields are of type
 * string.
 */
typedef struct {
    char nif[10]; ///< null-terminated 9-digit portuguese Tax
    identification numbers (TIN). Example "333333333\0"
    char nome[100]; ///< null-terminated person's name. Example
    "Cristiano Pedro Silva"
    char datanascimento[9]; ///< null-terminated 8-digit birth date
    (yyyymmdd). Example "20000624" for 24th of june 2000.
    char morada[100]; ///< null-terminated address: street and door
    number. Example "Praça 9 de Abril, 349" (UFP main address)
    char cod_postal[8]; ///< null-terminated 7-digit postal code.
    Example "4249004" for 4249-004 (UFP postal code)
    char freguesia[100]; ///< null-terminated district name. Example
    "Paranhos\0" (UFP district)
} pessoa;
```

A informação é carregada para a memória do sistema usando um ou vários ficheiros de texto¹. O sistema poderá, igualmente, gravar num ficheiro de texto os dados residentes em memória

Adicionalmente, cada pessoa registada no sistema poderá ter associadas a si pessoas que são familiares mais diretos, como por exemplo, cônjuge, filhos, irmãos, cunhados ou pais. O sistema pode representar, para cada pessoa, a lista dos familiares diretos associados, sem especificar qual o grau de parentesco envolvido. Essa informação é, igualmente, carregada para a memória do sistema usando um ou vários ficheiros de texto².

A aplicação a ser desenvolvida deverá suportar/implementar algoritmos de ordenação de strings, pesquisa binárias e pesquisa de strings. Deverá ainda permitir a geração de dados sintéticos na própria aplicação.

¹ Ver o exemplo "3persons.txt" fornecido em anexo.

² Ver o exemplo "3persons-family.txt" fornecido em anexo.

Juntamente com este texto é disponibilizado um conjunto de funções utilitárias para, por exemplo, permitir ler ficheiros de texto com informação de teste da aplicação.

2 Requisitos Funcionais

Não se pretende que desenvolva uma interface de utilizador completamente funcional. Pretende-se que desenvolva uma API de funções que satisfaçam os requisitos listados a seguir e funções de teste tal como descrito. De seguida descreve-se a lista dos requisitos a cumprir neste projeto de programação. O programa desenvolvido deverá permitir:

- R1. Fazer a ordenação das pessoas no sistema usando ordenação tipo string sorting (LSD e MSD). Deverá poder fazer a ordenação por nif, nome, data de nascimento ou código postal.
- R2. Deverá permitir, usando conectividade dinâmica, suportar a implementação de uma API union-find no conjunto de pessoas, usando a ligação familiares diretos para unir pessoas a outras pessoas e testar se duas pessoas estão ligadas por um caminho constituído por familiares diretos.
- R3. Fazer a pesquisa de pessoas nas estruturas ordenadas, por nif, nome, data de nascimento ou código postal, usando pesquisa binária.
- R4. Disponibilizar funções para inserir, remover, editar e listar pessoas na base de dados.
- R5. Listar todas as pessoas da base de dados (e escrever em ficheiro de texto).
- R6. Permitir a manipulação das estruturas de dados (e respectivas funcionalidades solicitadas) através da entrada e saída de dados via ficheiros de texto.
- R7. Correr cenários de teste (funções de teste) para todas as funcionalidades implementadas. Deverá implementar mecanismos / instrumentos de medição temporal da execução das principais funções de ordenação e pesquisa nesses testes. Esses cenários deverão incluir nos testes a leitura e escrita de ficheiros com dados. Os resultados dos testes deverão ser escritos para ficheiro de texto. Para cada teste realizado deverá ser explicitada (no ficheiro de texto) a função testada e o resultado do teste.

3 Documentação de AED1

3.1 Anotações e comentários no código fonte

As estruturas de dados e os algoritmos definidos devem ser implementados em C com os comentários apropriados, que facilitem a compreensão dos mesmos, escritos no código fonte desenvolvido. Todas as funções devem estar anotadas em formato doxygen³ incluindo uma explicação dos algoritmos implementados e uma menção ao desempenho dos mesmos assim como dos testes efetuados/implementados. As principais estruturas de dados e variáveis devem estar anotadas neste formato. Deverão usar o software doxygen para gerar a documentação com base nos comentários.

3.2 Relatório

Os alunos deverão entregar um relatório, um ficheiro de texto no formato .dox do doxygen⁴, que descreva as funcionalidades/requisitos implementados, parcialmente implementados e não implementados, mencionando sempre o número do requisito de acordo com este documento de especificação:

1. Funcionalidades implementadas: devem descrever de forma resumida todas as funções desenvolvidas para assegurar os requisitos funcionais solicitados.
2. Funcionalidades não implementadas: nesta secção devem identificar de forma resumida as funcionalidades não implementadas; devem ainda apontar-se as justificações/dificuldades que impediram o seu desenvolvimento.

4 Submissão

A aplicação (código) deve ser depurada de todos os erros de sintaxe. A aplicação final deve estar de acordo com os requisitos funcionais. Só serão considerados os programas que não contenham erros de sintaxe e implementem as funcionalidades pedidas. A documentação, em formato html (ou pdf), juntamente com todo o código-fonte desenvolvido, deve ser submetida num ficheiro zip/rar na plataforma de elearning.

³ (<http://www.doxygen.org>) A geração de anotações neste formato é parcialmente suportada pelo IDE codeblocks.

⁴ Ver o ficheiro “readme.dox” fornecido como exemplo.