

Trabalho 1 - Ordenação - CIX56

Paulo R. Lisboa de Almeida

1º Semestre - 2024

1 Descrição

Implemente os seguintes algoritmos em C:

Versões **recursivas** de: *Busca Sequencial* (ingênua), *Busca Binária*, *Insertion Sort*, *Selection Sort* e *Merge Sort*.

Versões **iterativas** de: *Busca Sequencial* (ingênua), *Busca Binária*, *Insertion Sort* e *Selection Sort*.

Juntamente com a implementação, um relatório em PDF com testes de tempo e quantidade de comparações deve ser entregue. A definição dos testes a serem feitos e discutidos no relatório ficam a cargo do aluno.

2 Template

Os algoritmos devem ser implementados utilizando como base o template disponibilizado no Moodle.

O arquivo *ordenacao.h* contém os protótipos (assinaturas) das funções que obrigatoriamente devem ser implementadas, no arquivo *ordenacao.c*.

Não é permitida a mudança nos protótipos de funções disponibilizados no arquivo *ordenacao.h*. É permitida a criação de funções auxiliares se necessário. Também é permitida a criação de arquivos de header (.h) e de implementação (.c) complementares se necessário.

É obrigatória a inclusão de um arquivo *main.c* com alguns testes nos algoritmos implementados. A definição dos testes fica a cargo do aluno.

O arquivo deve compilar sem erros ou avisos através do comando *make*.

O nome do binário gerado deve ser *trab1SeuGRR* (e.g., *trab1grr1234*). Ajuste isso no *makefile*. O binário deve executar em sistemas Linux executando o programa sem a passagem de nenhum parâmetro (e.g., *./trab1grr1234*).

3 Arquivos a serem entregues

Você deve incluir em um diretório compactado tar.gz (é obrigatório que o arquivo seja .tar.gz – arquivo *tarball* compactado via *Gzip*) de nome *trab1SeuGRR.tar.gz*. Se, por exemplo, seu GRR é 1234, o diretório contendo os arquivos deve se chamar *trab1grr1234*.

Compacte esse diretório, sendo que a versão compactada vai se chamar `trab1grr1234.tar.gz`. O diretório deve conter o seguinte:

- arquivos de código fonte `.c` e `.h` do programa;
- `makefile`;
- relatório em formato PDF.

Não inclua quaisquer outros arquivos irrelevantes para a compilação do projeto. Por exemplo, não inclua binários compilados, ou arquivos objeto (`.o`). A inclusão de arquivos irrelevantes pode acarretar em descontos de nota.

4 Relatório

Realize testes, e discorra sobre esses testes no relatório. Relacione, por exemplo, o tempo de execução em seu computador para diferentes algoritmos, e também o número de comparações realizadas pelos algoritmos.

Você pode incluir gráficos e figuras no relatório para torná-lo mais informativo e interessante.

O relatório deve ter no máximo duas páginas se utilizado espaçamento simples e coluna dupla, ou no máximo três páginas para espaçamento 1,5 ou duplo e formato de uma coluna.

Você pode ver um exemplo de relatório com template em <https://pt.overleaf.com/read/kfzrvbpnpnth>.

5 Entrega

O trabalho deve ser entregue via Moodle. A data limite para o envio está estipulada no link de entrega do Moodle.

Não serão aceitas entregas em atraso, exceto para os casos explicitamente amparados pelas resoluções da UFPR.

6 Grupos, Pesos e Datas

Grupos: trabalho individual.

Valor: 20% da nota do semestre.

Submissão: Via Moodle. Veja a data limite no link de submissão do Moodle.

7 Dicas

7.1 Contando o tempo

Para contar o tempo gasto por determinada função em C, inclua a biblioteca `time.h`. Veja a seguir um exemplo para contar o tempo em segundos necessário para se executar uma função.

```
#include <time.h>

//...
clock_t start, end; //variáveis do tipo clock_t
double total;

start = clock(); //start recebe o "ciclo" corrente
minhaFuncao(); //chama a função que desejamos medir o tempo
end = clock(); //end recebe o "ciclo" corrente
//o tempo total é a diferença dividida pelos ciclos por segundo
total = ((double)end - start)/CLOCKS_PER_SEC;
//total agora possui o tempo em segundos
printf("Tempo total: %f", total);
```

7.2 Função clock

A especificação em C da função `clock()`¹ é clara em dizer que a função deve retornar o tempo de CPU (*CPU time*)², mas algumas fontes citam que o Windows ignora a especificação e conta o tempo de parede (*Wall clock time*)³. Para simplificar, será aceito tanto o *CPU time* quanto o *Wall Clock time*, mas deixe claro em seu relatório qual sistema operacional você usou.

Para tornar os testes mais confiáveis, execute os programas com “tudo fechado” em sua máquina. Por exemplo, não abra o navegador web enquanto executar os testes, já que o navegador pode competir por recursos, e gerar diferenças nos tempos, especialmente para o *wall clock time*.

8 Distribuição da Nota

A nota do trabalho é composta por 70% referente a completude e qualidade de implementação, e 30% referente ao relatório. Esses valores podem variar para casos específicos. Por exemplo, caso um aluno entregue apenas o relatório e não a implementação, o relatório não terá direito aos 30% de nota.

Alguns descontos padrão, considerando uma nota entre 0 e 100 pontos para o trabalho:

- Plágio: perda total da pontuação para todos os envolvidos. Isso é válido mesmo para casos onde o plágio se refere a apenas um trecho do código.
- Algoritmos não recursivos ou não iterativos (dependendo da versão solicitada), ou implementados em linguagem diferente de C, serão desconsiderados.
- Não submissão via Moodle acarreta na perda total dos pontos.
- Falta de algum arquivo requisitado: desconto de 10 a 100 pontos.
- Inclusão de arquivos desnecessários (lixo): desconto de 5 a 20 pontos.
- Erros e avisos de compilação: desconto de 5 a 100 pontos.

¹ Ver Seção 7.23.2.1 de <<http://www.open-std.org/jtc1/sc22/WG14/www/docs/n1256.pdf>>

² <https://en.wikipedia.org/wiki/CPU_time>

³ <https://en.wikipedia.org/wiki/Elapsed_real_time>

- Nomes de arquivo incorretos: 5 pontos por arquivo.
- Arquivo com formato incorreto: 5 pontos por arquivo.
- Más práticas de programação (ex.: código obscuro, tipos incorretos de variáveis, má indentação, ...): 5 a 50 pontos.
- Relatórios mal formatados, com erros de português, ou que falham em apresentar os resultados de maneira satisfatória serão desconsiderados.

9 Demais Regras

- Dúvidas ou casos não especificados neste documento podem ser discutidos com o professor até a **data de entrega do trabalho**. Não serão aceitas reclamações após a data da entrega.
- Os alunos podem (e devem) procurar o professor, ou seus colegas de classe para tirar dúvidas quanto ao trabalho.
- O descumprimento das regras dispostas nesse documento podem acarretar na perda parcial ou total da nota do trabalho.
- Os trabalhos não serão aceitos após a data/hora limite.