Trabalho 2 - Ordenação - CIX56

Paulo R. Lisboa de Almeida

1º Semestre - 2024

1 Descrição

Implemente os seguintes algoritmos utilizando a linguagem C: *Merge Sort*, *Quicksort* e *Heapsort*. Para cada algoritmo, deve ser implementada uma versão que usa recursão (como feito em sala), e uma versão onde a recursão foi removida.

Para o algoritmo Heapsort, considere a funçao max-heapify. A versão dada em sala é a versão recursiva. Na versão não recursiva do Heapsort, remova a recursão de max-heapify.

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 *trab2SeuGRR* (e.g., *trab2grr1234*). Ajuste isso no *makefile*. O binário deve executar em sistemas Linux sem a passagem de nenhum parâmetro (e.g., *./trab2grr1234*).

3 Formato dos mallocs

Quando precisar realizar um malloc, o seguinte formato é obrigatório:

tipo val = (tipo*) malloc(QTDE* sizeof(tipo));*

Exemplo: *int* val = (int*) malloc(sizeof(int));*

O formato com o *cast* logo antes do malloc é obrigatório devido aos compiladores e flags que serão usados durante os testes. Para saber mais, leia <https://prlalmeida.com.br/knowledge-base/malloc>.

4 Pilha

Ao remover a recursão, pode ser necessária uma estrutura do tipo pilha em C. Você pode criar seu próprio algoritmo de pilha, ou então usar a estrutura de pilha de alguma biblioteca pronta. Durante a execução dos testes, **exclusivamente para a versão não recursiva dos algoritmos**, é garantido que a **quantidade das chamadas** empilhadas não excederá 1.048.576 chamadas.

Caso você use alguma biblioteca externa para pilha, garanta que essa biblioteca está disponível nas instalações padrão do DInf. Durante os testes não serão instaladas bibliotecas extras nos computadores, e falhas de compilação podem acarretar na perda total dos pontos do trabalho.

5 Arquivos a serem entregues

Você deve criar um diretório compactado tar.gz (arquivo *tarball* compactado via *Gzip*) de nome trab2SeuGRR.tar.gz. Se, por exemplo, seu GRR é 1234, o diretório contendo os arquivos deve se chamar trab2grr1234. Compacte esse diretório, sendo que a versão compactada vai se chamar trab2grr1234.tar.gz. O diretório deve conter o seguinte:

- arquivos de código fonte .c e .h do programa.
- makefile.

Não inclua quaisquer outros arquivos irrelevantes. Por exemplo, não inclua binários compilados, ou arquivos objeto (.o), sob pena de descontos de nota.

6 Entrega, Grupos, Pesos e Datas

O trabalho é individual e tem peso de 20% no semestre. A submissão é via Moodle. Veja a data limite no link de submissão do Moodle. Não serão aceitas entregas em atraso.

7 Distribuição da 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 que não foram implementados em C serão desconsiderados.
- Não submissão via Moodle acarreta na perda total dos pontos.
- Falta de algum arquivo requisitado: desconto de 10 a 100 pontos.
- Erros crassos, como vazamentos de memória, ou uso de variáveis globais: desconto mínimo de 20 pontos.
- Más práticas de programação, ou não seguir as diretivas do trabalho: 5 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.
- Nomes de arquivo incorretos: 5 pontos por arquivo.
- Arquivo com formato incorreto: 5 pontos por arquivo.

8 Demais Regras

- Dúvidas ou casos não especificados neste documento podem ser discutidos até a **data de entrega do trabalho**.
- 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.