

Primeiro Trabalho de Implementação

Algoritmos e Estrutura de Dados II

Segundo Semestre de 2013

O objetivo deste trabalho é implementar diferentes variações do Quicksort. Feita a implementação, uma bateria de testes será executada pelos professores. O objetivo da bateria de testes é comparar as variações do Quicksort baseadas em duas medidas de desempenho: **número de comparações entre elementos do vetor** e o **número de trocas de elementos do vetor**. Para isso, serão fornecidas duas funções, uma de comparação e outra de troca, que incluem contadores para as respectivas medidas de desempenho e que devem necessariamente ser usadas nas implementações.

Variações do Quicksort

Variação 1 - Quicksort Simples: este é o Quicksort tal como apresentado em sala de aula.

Variação 2 - Quicksort com Mediana: esta variação do Quicksort escolhe o pivô para partição como sendo a mediana dentre **três** (3) elementos do vetor, escolhidos aleatoriamente de maneira uniforme.

Variação 3 - Quicksort com Inserção m : esta variação do Quicksort utiliza o algoritmo de Inserção quando o tamanho das partições (isto é, pedaços do vetor) for menor ou igual a m . Ou seja, a base da recursão é para pedaços do vetor com tamanho menor ou igual a m , quando deve ser usado a ordenação por Inserção para ordenar esses pedaços.

Na bateria de testes, serão avaliados **dois** *recursos computacionais*:

1. o número de comparações entre elementos do vetor,
2. o número de trocas de elementos do vetor.

Para isso, são fornecidas duas funções, **compara** e **troca**, implementadas dentro do arquivo `biblioteca.c`. Essas funções devem ser usadas nas implementações para a correta avaliação do trabalho. Para exemplificar o uso dessas funções, é fornecido o arquivo `selecao.c` que contém a implementação completa da Ordenação por Seleção.

Detalhes de Implementação

Serão fornecidos arquivos de código fonte de dois tipos:

1. Arquivos de código fonte “esqueleto” que devem ser preenchidos com sua implementação,
2. Arquivos de header, exemplo ou auxiliares.

Os arquivos que **devem** ser preenchidos (fazem parte da avaliação) são:

`quicksort.c` : a implementação do Quicksort Simples (Variação 1)

`quicksort-mediana.c` : a implementação do Quicksort com Mediana (Variação 2) e a função respectiva para calcular a mediana de três valores.

`quicksort-insercao.c` : a implementação do Quicksort com Inserção m (Variação 3)

`particiona.c` : a implementação do Particiona (usado nas três variações)

`insercao.c` : a implementação da Ordenação por Inserção (usado como subrotina da Variação 3)

Os cabeçalhos das funções, os arquivos de header (`.h`) e o arquivo `biblioteca.c` **não devem ser modificados**, pois essas funções serão usadas na bateria de testes que o programa será submetido. Tais arquivos são:

`selecao.c` : o código-fonte da Ordenação por Seleção (só usado para exemplificar o uso das funções de troca e comparação).

`biblioteca.c` : esse arquivo contém o código-fonte que deve ser usado das funções de comparação e troca, os contadores para tais recursos computacionais, e a função `sorteia`, que deve ser usada no Quicksort com Mediana para fazer a seleção aleatória dos pivôs. Além disso, contém funções auxiliares para zerar os contadores e retornar o número total de trocas e comparações.

`teste.c` : um exemplo de bateria de testes que será executada. É **altamente recomendado** fazer seu programa funcionar para essa bateria de testes, mas sabendo que a bateria de teste que o programa será submetido pode ser trocada.

É importante, na implementação de cada variação, **não** esquecer de usar as funções fornecidas de troca e comparação!

O trabalho deve ser feito de forma que possa ser compilado e executado nas servidoras de computação do Departamento de Informática.

O que você deve entregar

São **cinco** arquivos que devem ser entregues:

- `quicksort.c`
- `quicksort-mediana.c`
- `quicksort-insercao.c`
- `particiona.c`
- `insercao.c`

Esses arquivos devem ser compactados num arquivo `.tar.gz` (detalhes abaixo).

Forma de Entrega

O trabalho pode ser feito em grupos de até dois alunos.

Os arquivos devem ser empacotados em um arquivo `grr1-grr2.tar.gz`, onde `grr1-grr2` é uma string com os GRR's dos integrantes da equipe. Ao descompactar este arquivo deverá ser criado um diretório de nome `grr1-grr2` que conterà todos os demais arquivos.

Este arquivo deve ser enviado como anexo por e-mail ao endereço do professor (vignatti, andrey @ inf) com o assunto "CI056-trab1" (**exatamente**).

Data de Entrega

O trabalho pode ser entregue até às 23h59m do dia 06/novembro. Entregas feitas após esse prazo **não serão** avaliadas (recebem nota zero).

Critério de Avaliação

Os critérios de avaliação são os seguintes:

- O trabalho deve ser entregue no formato que foi especificado, assim como explicado na Seção “Forma de Entrega”.
- O trabalho deve compilar e executar sem problemas, inclusive sendo compilado com `Makefile` e podendo ser testado com o arquivo `teste.c`.
- O trabalho está correto, ou seja, as implementações das variações do Quicksort realmente ordenam e estão implementadas de acordo com o especificado.
- Clareza do código.