

Estrutura de Dados 2

Roteiro de Laboratório 5 – *Merge sort*

1 Objetivo

O objetivo deste laboratório é implementar uma série de variantes do *merge sort* e fazer uma análise empírica (experimental) do seu desempenho.

2 Configuração dos experimentos

Utilize o arquivo `ED2_Lab05_in.tar.bzip2` disponibilizado no AVA para testar os programas que você desenvolver neste laboratório. Este arquivo possui entradas de 100K, 1M e 10M inteiros para ordenação. As variantes da entrada são as mesmas usadas anteriormente: aleatório, ordenada, ordenada reversa e parcialmente ordenada.

IMPORTANTE: Extraia o conteúdo do arquivo em um diretório local do seu computador, por exemplo em `/tmp/`, para evitar o tráfego de arquivos grandes pela rede. Use, por exemplo, os comandos abaixo:

```
$ cat /tmp/ED2_Lab05_in.tar.bzip2_part_a /tmp/ED2_Lab05_in.tar.bzip2_part_b >
    /tmp/ED2_Lab05_in.tar.bzip2
$ tar -xvf /tmp/ED2_Lab05_in.tar.bzip2
```

Utilize o programa cliente desenvolvido no Exercício 2 do Laboratório 3 para realizar os testes dos algoritmos de *merge sort*. Meça o tempo de execução de cada uma das variantes do *merge sort* para cada uma das entradas de teste.

3 Variantes do *merge sort*

Implemente, teste e meça o tempo de execução de cada uma das versões do *merge sort* abaixo:

- **Versão 1:** *merge sort* clássico *top-down* recursivo sem nenhuma otimização. (Veja os slides 7 e 8 da Aula 05.)
- **Versão 2:** *merge sort top-down* recursivo com *cut-off* para *insertion sort*. Implemente o seu código de forma que seja fácil modificar o valor de *cut-off*. (Veja o slide 15 da Aula 05.) Varie o valor de *cut-off* a partir de 1 e determine o valor ideal para a sua implementação.
- **Versão 3:** *merge sort top-down* recursivo com *merge skip*. (Veja o slide 16 da Aula 05.) Implemente essa versão a partir da Versão 1, isto é, não use *cut-off*. Assim, comparando separadamente as versões você pode determinar o ganho individual de cada otimização.
- **Versão 4:** fusão das Versões 2 e 3, isto é, usar as duas otimizações: *cut-off* e *merge skip*.
- **Versão 5:** *merge sort bottom-up* sem nenhuma otimização. (Veja o slide 20 da Aula 05.)
- **Versão 6:** altere a Versão 5 para implementar o que seria o “*cut-off*” na versão *bottom-up*.
- **Versão 7:** altere a Versão 6 para implementar o *merge skip* na versão *bottom-up*.

Faça uma análise das 7 versões do algoritmo segundo o desempenho. Há alguma que se destacou? Qual versão você escolheria e por que?