

Trabalho em Grupo 2

Ordenação externa

Considere o seguinte método de ordenação por intercalação externa (external merge sort) para ordenar N registros de um arquivo utilizando memória principal com capacidade para armazenar e ordenar no máximo M ($M < N$) registros simultaneamente:

- 1) Ler até M registros do arquivo de entrada e ordená-los utilizando um algoritmo de ordenação convencional (Quicksort, Mergesort, Heapsort, etc);
- 2) Escrever os registros ordenados em um arquivo intermediário;
- 3) Repetir os passos 1 e 2 até que os N registros sejam lidos e gravados em arquivos intermediários e ordenados;
- 4) Para cada par de arquivos intermediários ordenados, realizar intercalação dos registros e escrever os registros ordenados em um novo arquivo intermediário;
- 5) Executar o passo 4 enquanto esse passo gerar mais de um arquivo intermediário.

Execute a sua implementação para o arquivo de entrada fornecido ou crie outro alternativo, calculando os tempos de execução do programa para cada entrada. Além disso, para cada arquivo de entrada, calcule a quantidade de vezes que cada registro é lido e escrito no disco. Faça um relatório com gráficos que comparam o desempenho do método implementado para cada entrada. Descreva os detalhes dos experimentos realizados e discuta sobre os resultados observados. Para este trabalho, considere $M = 5.000$.

Formato de entrega

Arquivo comprimido (nome do arquivo: nome1_nome2_nome3_trab3.zip) contendo os seguintes arquivos:

- i) Relatório em PDF
- ii) Código fonte do programa

Entrada

Cada arquivo de entrada contém N ($10.000 \leq N \leq 1.000.000$) linhas. Cada linha i contém um número inteiro a_i ($0 \leq a_i \leq 10.000.000$).

Arquivos de entrada sugeridos em anexo.

Saída

O seu programa deve gerar um arquivo contendo os N números do arquivo de entrada em ordem não-decrescente.

Tamanho dos Grupos

Até 3 alunos.