

3º Exercício: Ordenação

Comparar três métodos de ordenação: BubbleSort, MergeSort e QuickSort com relação à eficiência de tempo.

----- Parte A

Dado um arquivo contendo nomes de pessoas, colocar os nomes em ordem alfabética. Escreva um único programa que utiliza os 3 métodos e gera 3 arquivos de saída. Cada saída deve conter o tempo de execução e o número de comparações feitas. Detalhes:

Cada nome de pessoa: no máximo 50 letras.

Quantidade de pessoas: sem limite ☺

Arquivo entrada:

C:\\Lab3\\entrada3.txt

Arquivos de saída:

C:\\Lab3\\seunome3bubble.txt

C:\\Lab3\\seunome3merge.txt

C:\\Lab3\\seunome3quick.txt

-----Parte B

Você deve rodar os 3 métodos com diversas entradas geradas randomicamente e elaborar tabelas e gráficos comparando-os.

Entregue um relatório contendo:

Tabela: tamanho da entrada / número comparações / tempo

- fazer uma tabela para cada método.

Gráficos para o número de comparações

- fazer um gráfico para BubbleSort

- fazer um gráfico para MergeSort e QuickSort juntos

Gráficos para o tempo de execução

- idem acima

O tipo deve ser “gráfico de dispersão com linhas” ou tipo análogo.

Análise dos resultados. ☺

Detalhes:

Cada tabela e cada gráfico devem ter no mínimo 15 “pontos”.

Use somente entradas com tamanho múltiplo de 1000.

Podem ser só múltiplos de 2000, de 5000, etc... Você decide ☺

Testar cada método até o tamanho máximo que conseguir rodar em 2 segundos, lembrando que esses tamanhos podem ser diferentes para cada método.

Sempre que “possível”, rodar MergeSort e QuickSort com as mesmas entradas! ☺

Devem-se usar os mesmos BubbleSort, MergeSort e QuickSort da parte A,

Mas podem estar em diferentes arquivos fonte (.CPP).

Inicialmente não é necessário entregar os programas nem as entradas usadas na parte B.

Mas guarde-os. Pois, durante a correção, podem ser solicitados para conferência.

-----Importante

Os algoritmos MergeSort e QuickSort devem ser os **mesmos** vistos em aula, mas com as adaptações necessárias para ordenar vetor de strings.

Número de comparações: contabilizar somente comparações entre strings.

Tempo gasto: cuidado para NÃO contar o tempo inicial para preencher o vetor nem o tempo final para escrever no arquivo.

Fazer individualmente.

Significa não ver os relatórios nem os resultados dos outros alunos.

Você receberá, disponível no TIDIA:

- códigos do MergeSort e QuickSort vistos em aula
- exemplo de código para marcar tempo de execução
- programa que gera strings randomicamente.

Observações:

Posso usar os arquivos gerados pelo meu colega?

- Não.

Posso usar um gerador de strings feito pelo meu colega?

- Sim.

No relatório, dizer:

“Usei um gerador feito pelo aluno Fulano” ou

“Usei o gerador fornecido no TIDIA” ou

“Fiz meu próprio gerador...”, etc ☺

-----Entregar pelo TIDIA:

Referente à parte A:

seunome3.cpp

seunome3.exe

Referente à parte B:

seunome3relatório.doc ou .pdf