

Especificação do Trabalho

Objetivo

O trabalho tem como objetivo consolidar o conhecimento sobre as características do problema de ordenação externa. Neste contexto, o estudante deverá entregar o código-fonte, livre de erros para que seja possível compilá-lo e executá-lo de acordo com a presente especificação. O programa entregue deve ser capaz de ordenar um arquivo de entrada levando em consideração restrições de tamanho memória interna e números de arquivos a serem manipulados simultaneamente.

Sobre o método de ordenação

O método a ser considerado no trabalho é o conhecido como **ordenação balanceada de p -caminhos** (*p -way merge sort*). O parâmetro p restringe o número de arquivos a serem manipulados concomitantemente a $2p$. Para efeito da implementação solicitada, deve-se assumir que a memória principal não é capaz de conter mais que p registros. Além disso, o método de ordenação deve usar o algoritmo de **seleção por substituição** para gerar as sequências ordenadas iniciais. Similarmente, a intercalação de até p sequências ordenadas deve ser baseada em *heap* mínima.

Sobre a interface de entrada e saída

Seu programa deve ser executado em linha de comando (*prompt shell*), recebendo três parâmetros de entrada: o valor de p , o nome do arquivo de entrada e o nome no arquivo de saída. Os registros a serem ordenados no arquivo de entrada serão números inteiros, um por linha.

Como ilustração, suponha que o nome do executável é `pway`, o nome do arquivo de entrada é `input.txt`, do arquivo de saída é `output.txt` e $p = 2$. Na linha de comando, deve-se digitar:

```
pways 3 input.txt output.txt
```

Observe que $p < 2$ não uma opção válida.

Como saída, a execução do programa, além de fornecer os registros do arquivo de entrada ordenados, ele deve exibir como saída três valores: o número de registros processados; p ; o número de sequências iniciais geradas; e o número de passagens para se obter uma única sequência ordenada. Por exemplo, suponha que o arquivo de entrada `input.txt` contém os seguintes 25 registros

18 7 3 24 15 5 20 25 16 14 21 19 1 4 13 9 22 11 23 8 17 6 12 2 10

Então, as seguintes saídas devem ser exibidas para em função do valor de p usado:

```
./pways 3 input.txt output.txt
#Regs Ways #Runs #Parses
25      3  5      2
./pways 2 input.txt output.txt
#Regs Ways #Runs #Parses
25      2  7      3
./pways 4 input.txt output.txt
#Regs Ways #Runs #Parses
25      4  4      1
```

Considerações e critérios

- A implementação não deve usar métodos de ordenação em memória interna em nenhuma fase da ordenação externa. Além disso, arquivos de entrada e saída temporários devem ser usados. Qualquer manipulação de arquivos como sequências em memória interna não será permitida. Não atender tais restrições ou usar de plágio implicará em nota zero no trabalho.
- O trabalho é individual e valerá uma nota de 0 a 10. Tal nota será usada para alterar a menor nota P de uma das provas, substituindo-a por $(P + T)/2$ contanto que $P < T$. A entrega do trabalho é de caráter opcional.
- Os arquivos de entrada e saída podem ser grandes. Portanto, o programa não deve conter qualquer restrição com relação ao número de registros no arquivo de entrada.
- O código deve ser entregue de forma que sua correção possa ser realizada independente da máquina hospedeira. Para tanto, serviços em nuvem podem ser usados desde que a autorização de acesso esteja liberada ao professor.
- Trabalhos que contenham erros de compilação não serão corrigidos. A linguagem de programação a ser utilizada é de livre escolha entre C, C++, Java ou Python. Outras linguagens podem ser permitidas, desde que previamente autorizadas.
- O trabalho deve ser entregue em 30 dias a partir da data desta divulgação.