DCC–IME–USP

Routo Terada
Sala 204C Tel.: 3091-5918
e-mail rt@ime.usp.br

Introdução à Computação II

Curso de C. Moleculares – Primeiro Semestre de 2004

Exercício-Programa 2, Peso 1 Data de entrega: até **15 de junho de 2004.**

Ordenação por Concatenação (QuickSort)

Este exercício é para ser elaborado em linguagem C. O objetivo é corroborar a complexidade de tempo $O(n)$ do algoritmo QuickSort, se os dados de entrada são distribuídos uniformemente.

As funções solicitadas abaixo podem possuir outros parâmetros de entrada ou de saída.

1. Escrever uma função – Mediana(a, b, c) – que calcula a mediana de três elementos inteiros a, b, c.
2. Escrever uma outra função – Particiona(M, n) – que particiona um vetor M de n inteiros de acordo com a mediana dos elementos em $M[0], M[j], M[n-1]$ onde j é o índice do "meio" de M (Página 47 do livro)
3. Escrever uma função – OrdConc(M, n) – que ordena um vetor M de n inteiros (Página 49 do livro)
4. Escrever uma função - GeraVetor(M, n, a, b) – que gera um vetor M de n inteiros distribuídos uniformemente no intervalo $[a, b]$, utilizando a função $rand()$ e $srand()$ da biblioteca `stdlib.h`

Escrever um programa que utilize necessariamente as funções solicitadas, e calcula uma estatística de quantas comparações $NCompar$ entre elementos e quantas trocas de posição $NTrocas$ entre elementos do vetor M são efetuadas pelas três funções Mediana, Particiona e OrdConc no **total**, para cada vetor M.

O intervalo $[a, b]$ deve ser $[0, 100]$. Você deve gerar $NvetorM$ vetores M, sendo cada vetor M de comprimento $NComprM$. Os seus testes deverão ser para os valores $NvetorM = 50$ e $NComprM = 200$. A memória para armazenar um vetor M de comprimento $NComprM$ deve ser reutilizada para o vetor gerado seguinte..

Após a ordenação de cada vetor, o programa deve:

1. mostrar na tela os primeiros 50 elementos do vetor, em ordem crescente (com pausa apropriada após um número adequado de linhas, para facilitar a leitura dos dados na tela)
2. os valores $NCompar, NTrocas$ para este vetor. Observe que não há necessidade de armazenar estes valores para o cálculo da média e do desvio padrão solicitados a seguir.

O programa não possuirá qualquer dado de entrada ou qualquer arquivo de saída em disco. Após uma única execução do programa, a saída final na tela deverá conter pelo menos:

1. os valores $NvetorM, NComprM$ utilizados
2. a **média** e o **desvio padrão** dos $NvetorM$ valores de $NCompar$ e $NTrocas$.

Observações:

1. A nota atribuída ao exercício-programa também vai depender dos seguintes fatores: qualidade da decomposição do programa em funções e procedimentos; qualidade e clareza dos algoritmos; qualidade das estruturas de dados; qualidade da listagem; qualidade do layout, da ortografia e da pontuação; qualidade dos nomes das variáveis e tipos. Exercícios-programas com erro de sintaxe receberão nota zero.
2. Este exercício é para ser feito *individualmente*.
3. Entregue todos os arquivos relevantes "zipados" num único arquivo para o servidor PANDA (panda.ime.usp.br).
4. Coloque comentários em seu programa explicando o que cada etapa do programa significa! Isso será levado em conta na sua nota.
5. Coloque como comentário o seu nome, número USP, qual o compilador, qual o sistema operacional e qual o modelo de computador que V usou.
6. Faça uma saída clara! Isso será levado em conta na sua nota.
7. Não deixe para a última hora. Planeje investir 70 porcento do tempo total de dedicação em escrever o seu programa todo e simular o programa SEM computador (eliminando erros de lógica) ANTES de digitar e compilar no computador. Isso economiza muito tempo e energia.