4ª Lista de Exercícios

Estrutura de Dados

Prof. Hamilton José Brumatto

Estrutura de Dados

1. Considere um problema de geometria computacional, onde um ponto é determinado pelas suas coordenadas (x, y). Neste contexto há a necessidade de algumas informações:
   (a) Qual a distância entre 2 pontos?
   (b) Quando 3 pontos são colineares?
   (c) 3 pontos formam um triângulo? Qual a área?
   (d) E o mais divertido: Dado um conjunto de n pontos, quais são os dois mais próximos?

   Construa um código que crie uma estrutura para representar um ponto e funções para resolver cada uma das questões acima.
2. Ainda em geometria computacional, outra estrutura interessante é uma circunferência, que é representada por um ponto (x, y), o seu centro, e pelo seu raio r. Crie uma estrutura para representar uma circunferência e funções que retornem as seguintes informações:
   (a) Se duas circunferências têm interseções.
   (b) Se duas circunferências são concêntricas.
   (c) Se uma circunferência abrange completamente outra.
3. Considere um sistema linear de equações de, no máximo, 6 incógnitas. Construa um programa que receba os coeficientes e termos independentes de cada equação e retorne, quando há solução, o valor das variáveis que tornam o sistema verdadeiro.
4. Em uma programação de jogos, usa-se muita estrutura para representar os diversos elementos do jogo: jogador, ambiente, objetos de jogo, ... Considere um jogo simples de tênis (aquele antigo do "tele-game"). Pede-se:
   (a) Como você representaria um jogador (raquete), a bola, a quadra?
   (b) Que ações precisam ser programadas para que o jogo funcione? (quem interage com quem?)
5. Em computação distribuída, por ventura é necessário que um dos processos participantes se declare como líder. Existem vários algoritmos para "*eleição de líder*", um algoritmo simples se dá da seguinte forma:
   (a) Os nós estão organizados em uma lista circular, e cada processo possui um ID (todos distintos)
   (b) Um processo qualquer inicia a eleição.
   (c) Durante a eleição, um processo recebe um ID de seu processo vizinho anterior e compara com o seu próprio ID:
      i. Se o ID recebido for menor que o seu, ele envia o seu próprio ID para o próximo.
      ii. Se o ID recebido for maior que o seu, ele encaminha o ID recebido para o próximo.
      iii. Se o ID recebido for o seu próprio, então ele é o líder. (E todos os demais sabem que não são, por quê?)

Pede-se, usando uma lista encadeada circular, simule uma eleição de líder, inserindo antes, alguns processos com IDs aleatórios na lista.

6. Em uma lista duplamente encadeada circular, é possível caminhar pela lista em ambos sentidos. O jogo de eliminação ficaria mais interessante, se em cada lance de jogada a contagem fosse num sentido diferente. Implemente tal jogo.
7. Considere o seguinte algoritmo de ordenação por troca: MERGE_SORT.

MERGE_SORT

Entrada: Uma sequência de números
Saída: A mesma sequência de números, ordenada

Algoritmo MERGE_SORT(A,p,r)
  se $p < r$ **então**
    $q \leftarrow (p + r)/2$
    MERGE_SORT(A, p, q)
    MERGE_SORT$(A, q + 1, r)$
    INTERCALA(A, p, q, r)

A funçao INTERCALA é representada abaixo:

INTERCALA

Entrada: Uma sequência onde no intervalo [p,r] a sequência está ordenada no subarranjo [p,q] e no subarranjo [q+1,r].
Saída: A mesma sequência, ordenada no intervalo [p,r].

Algoritmo INTERCALA(A,p,q,r)
  para $i \leftarrow p$ **até** q **faça**
    $B[i] \leftarrow A[i]$
  para $j \leftarrow q + 1$ **até** r **faça**
    $B[r + q + 1 - j] \leftarrow A[j]$
  $i \leftarrow p$
  $j \leftarrow r$
  para $k \leftarrow p$ **até** r **faça**
    se $B[i] \leqslant B[j]$ **então**
      $A[k] \leftarrow B[i]$
      $i \leftarrow i + 1$
    senão
      $A[k] \leftarrow B[j]$
      $j \leftarrow j - 1$

Pede-se: Gere uma sequência de valores inteiros em um vetor e aplique o algoritmo para ordená-los.