Segunda Prova de Algoritmos e Estruturas de Dados I
18/10/2006

Perguntas comuns e suas respostas:

- P: Tenho uma dúvida na questão tal.
  R: A compreensão do enunciado faz parte da prova.
- P: Se eu consultar algum material próprio ou de algum colega, o que acontecerá comigo?
  R: A prova é individual e sem consulta. Qualquer tentativa de fraude acarretará abertura de processo administrativo na UFPR.
- P: Posso fazer a prova a lápis?
  A prova é um documento, portanto deve ser feita à caneta.
- P: O que será corrigido?
  R: A lógica, a criatividade, a sintaxe, o uso correto dos comandos, a correta declaração dos tipos, os nomes das variáveis, a indentação, uso equilibrado de comentários no código e, evidentemente, a clareza.

Questões (Valores entre parênteses. Total 100 pontos):

1. (50 pontos) Escreva uma função em Free Pascal que substitui em um texto a primeira ocorrência de uma palavra por outra. A função deve retornar `true` se a substituição for bem sucedida e `false` caso a palavra não seja encontrada no texto. O texto e as palavras são representados por vetores do tipo `char`. Por exemplo:

```
        +---+---+---+---+---+---+---+---+---+---+
texto1  | e | x | e | m | p | r | o |   | u | n |
        +---+---+---+---+---+---+---+---+---+---+

          +---+---+---+---+---+
palavra1  | r | o |   | u | n |
          +---+---+---+---+---+

          +---+---+---+---+---+---+---+
palavra2  | l | o |   | d | o | i | s |
          +---+---+---+---+---+---+---+

        +---+---+---+---+---+---+---+---+---+---+---+---+
texto2  | e | x | e | m | p | l | o |   | d | o | i | s |
        +---+---+---+---+---+---+---+---+---+---+---+---+
```

   A função recebe como parâmetros o texto, a palavra a ser substituída e a nova palavra. No exemplo, `texto1` mostra o estado inicial do texto e `texto2` o estado do texto após a substituição da `palavra1` pela `palavra2`.

   Você pode usar, caso seja necessário, a função:

```
buscachar(texto, pos, letra);
```

   que busca um caractere (`letra`) a partir de uma determinada posição (`pos`) em um vetor que contém o texto (`texto`). A função `buscaletra` retorna a posição no vetor `texto` da primeira ocorrência de `letra`, se `letra` não aparece no texto a função retorna `-1`.

2. (50 pontos) Um *algoritmo genético* é um procedimento computacional de busca, inspirado no processo biológico de evolução, que otimiza a solução de um problema. O problema é modelado por: uma população de indivíduos que representam possíveis soluções; uma função que avalia a qualidade da solução representada por cada indivíduo da população e um conjunto de operadores genéticos. Os indivíduos são dados por seqüências de genes que representam características da solução do problema. O procedimento consiste em aplicar os operadores genéticos sobre a população, gerando novos indivíduos e selecionar os mais aptos para constituirem uma nova população. Esse processo é repetido até que uma solução adequada seja obtida. Dentre os operadores genéticos, o mais importante é o de recombinação genética (*crossover*) de dois indivíduos. Esse operador corta em duas partes as seqüências de genes de dois indivíduos pais (`pai1` e `pai2`) e gera dois novos indivíduos filhos (`filho1` e `filho2`). `filho1` é dado pela contatenação da primeira parte dos genes de `pai1` com a segunda parte de `pai2` e `filho2` pela concatenação da primeira parte de `pai2` com a segunda parte de `pai1`. O diagrama abaixo exemplifica a operação em indivíduos representados por vetores de números inteiros onde a primeira posição contém o tamanho do vetor:

```
                     corte1
        +----+---+---+---#---+---+---+---+---+---+---+
pai1    | 11 | 1 | 1 | 1 # 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
        +----+---+---+---#---+---+---+---+---+---+---+

                             corte2
        +----+---+---+---+---+---#---+---+---+---+---+
pai2    | 10 | 3 | 3 | 3 | 3 | 3 # 4 | 4 | 4 | 4 | 4 |
        +----+---+---+---+---+---#---+---+---+---+---+

        +----+---+---+---+---+---+---+---+---+
filho1  |  8 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 |
        +----+---+---+---+---+---+---+---+---+

        +----+---+---+---+---+---+---+---+---+---+---+---+---+---+
filho2  | 13 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
        +----+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Escreva um procedimento em Free Pascal que execute a operação de recombinação descrita acima, usando a estrutura de dados proposta. O procedimento deve receber seis parâmetros, um vetor representando o primeiro pai, a posição de corte no primeiro pai, um vetor representando o segundo pai, a posição do corte no segundo pai, e dois vetores que receberão os novos indivíduos. No exemplo apresentado a chamada do procedimento seria:

```
corte1 := 4;
corte2 := 6;
crossover(pai1, corte1, pai2, corte2, filho1, filho2);
```

Note que os vetores devem iniciar na posição zero e essa posição é usada para armazenar o tamanho do vetor. No caso do exemplo, `pai1[0]=11`, `pai2[0]=10`, `filho1[0]=8` e `filho2[0]=13`. Os pontos de corte devem estar dentro dos vetores: `1 < corte1 <= pai1[0]` e `1 < corte2 <= pai2[0]`.