EXERCÍCIO PRÁTICO 3

Enunciado baseado no trabalho proposto na disciplina "Estruturas de Dados" da Universidade Estadual de Campinas (http://www.lrc.ic.unicamp.br/~luciano/courses/mc202-2s2009/lab02.pdf).

INSTRUÇÕES:

- A atividade deve ser realizada em duplas e entregue até às 23h59 do dia **19/03/2013** via Moodle, por apenas um integrante da dupla.
- A atividade deve ser entregue na forma de um arquivo compactado nomeado da seguinte forma nome1_nome2.zip ou nome1_nome2.rar, onde nome1 e nome2 são os nomes dos integrantes da dupla.
- Faça o exercício apenas com a sua dupla.
- Endentem o código, deêm nomes de variáveis que reflitam suas funções e façam comentários pertinentes no código.
- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais estará.
- Em caso de dúvida entre em contato por email alinemello@unipampa.edu.br ou pessoalmente na sala 229.

AVALIAÇÃO:

O EP3 vale 10 pontos e tem peso igual ao de uma prova, ou seja 1,67 na nota final da disciplina. A nota máxima do trabalho perde 1 ponto a cada dia de atraso na entrega. Por exemplo, caso o trabalho seja entregue no dia 20/03/2013, a nota máxima será 9.

Trabalhos desenvolvidos com **listas encadeadas circulares receberão 1 ponto extra**.

ENUNCIADO:

Uma matriz esparsa é aquela onde a maioria dos elementos possui o valor zero. Existem diversas aplicações que fazem uso de matrizes esparsas na engenharia e na matemática, como o método das malhas para resolução de circuitos elétricos ou sistemas de equações lineares. Essas matrizes também são comuns na computação, como as matrizes de adjacência de grafos, e mapas de bits (bitmap) em imagens digitais.

Para escrever programas que manipulem matrizes esparsas devemos ter certos cuidados quanto à memoria utilizada, visto que simplesmente armazenar estas matrizes em um array bidimensional não é eficiente. Por que alocar memória para guardar diversos elementos que valem zero? Neste caso, alternativas mais eficientes no que diz respeito ao armazenamento destas matrizes devem ser consideradas.

Neste exercício iremos utilizar matrizes esparsas para representar bitmaps de imagens digitais. Para isso, vamos construir uma lista encadeada para armazenar somente as posições da matriz onde os elementos diferem do valor zero. Fazendo isso, é possível economizar memória e, ao mesmo tempo, permitir que os elementos da matriz sejam modificados dinamicamente, inserindo ou removendo nós nesta lista.

Um bitmap é um tipo de organização de memória usado para armazenar imagens digitais. O nome tem origem em mapas de bits, fazendo referência a vetores de bits usados para representar os pixels da imagem. Existem diversos formatos de arquivos bitmap disponíveis, mas iremos trabalhar com o formato PGM[1] (Portable Gray Map) em sua versão básica, que apresenta uma estrutura bem simples para armazenar figuras em escala de cinza.

[1] http://netpbm.sourceforge.net/doc/pgm.html

Um arquivo PGM é um arquivo texto contendo somente caracteres ASCII, constituído das seguintes partes:

1. Um identificador do tipo do arquivo, seguido de uma quebra de linha;
2. As dimensões da matriz que contém os pixels da imagem, seguido de uma quebra de linha;
3. O maior valor da escala de cinza possível (Maxval) na figura, seguido de uma quebra de linha;
4. Os elementos da matriz de pixels com os valores de tons de cinza da imagem. Cada valor de cinza é um número entre 0 e o máximo valor possível Maxval, sendo o 0 a cor preta e o Maxval a cor branca.

Observações:

- Vamos usar sempre o identificador P2 para o formato do arquivo PGM.
- As dimensões da matriz representam o número de colunas e linhas, nesta ordem.
- O maior valor da escala de cinza que vamos usar é sempre 255.
- A ordem dos elementos na matriz é a mesma ordem dos pixels na imagem. Eles precisam estar ordenados da esquerda para a direita e de cima para baixo. Exemplo da ordem dos elementos para uma matriz 3x3: [1,1] [1,2] [1,3] [2,1] [2,2] [2,3] [3,1] [3,2] [3,3].
- Entre os elementos da matriz pode haver um ou mais espaços e/ou quebras de linha. A quantidade de espaços e/ou quebras de linhas não deve ser considerada na hora da leitura da imagem nem será avaliada durante a correção automática do Susy. Assim, você pode colocar todos os elementos da matriz em uma única linha; organizá-los da forma usual como no exemplo abaixo; ou colocar um único elemento por linha, como nos nossos arquivos de testes.
- É possível salvar este conteúdo em um arquivo texto com a extensão .pgm e abrir esta imagem em qualquer programa de manipulação de imagem com suporte ao formato PGM (por exemplo, o Gimp).

Exemplo de um arquivo PGM

```
P2
12 9
255
0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0
0   0   187 187 187 0   0   255 255 255 0   0
0   0   187 0   0   0   0   255 0   0   0   0
0   0   187 0   0   0   0   255 0   0   0   0
0   0   187 0   0   0   0   255 0   0   0   0
0   0   187 187 187 0   0   255 255 255 0   0
0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0
```

Neste exercício, o seu programa deverá receber como entrada uma imagem no formato PGM (que na verdade é um texto como o apresentado acima). Você deve fazer a leitura dessa imagem e armazenar a matriz usando uma lista encadeada que contenha somente os elementos cujo valor difere de zero. Cada nó da lista encadeada que será criada deve ser um struct contendo um ponteiro para o próximo elemento da lista, além de valores numéricos para armazenar o valor de cinza correspondente e a referência à posição do pixel na matriz da imagem (que pode ser o número da linha e coluna da matriz, por exemplo).

Após a leitura e a criação da lista encadeada, você irá processar digitalmente esta imagem com as duas operações a seguir, que devem ser executadas na ordem em que estão apresentadas:

Inserção de borda: Deve-se colocar uma borda de 3 pixels de largura na cor branca ao redor de toda imagem. Para isso é preciso preencher todos os elementos das três primeiras e da três ultimas linhas e colunas com o valor 255. Não se preocupe em "perder" as informações da imagem que existem nas posições onde a borda será inserida. A Figura 1 apresenta um exemplo desta operação. Note que, como a página é branca, é difícil perceber a existência da borda, mas acredite: ela está aí! A execução desta operação requer a inserção de diversos nós na lista encadeada existente. Sugerimos que seja criada uma função para realizar esse processo de inserção.

(a) Imagem original

(b) Imagem com borda

Inversão da imagem: Após a inserção da borda, vamos fazer uma inversão nos pixels da imagem. Para inverter um pixel é preciso subtrair o valor atual do maior valor possível (no nosso caso, 255). A função que descreve essa operação é $V_{novo} = 255 - V_{atual}$. A Figura 2 apresenta um exemplo desta operação. Observe que com a inversão das cores é possível visualizar a borda inserida na operação anterior.
A execução desta operação requer a inserção e exclusão de diversos nós da lista ligada existente. Novamente sugerimos o uso de funções para realizar esse processo de remoção.

(a) Imagem com borda

(b) Imagem invertida

Ao final do processamento da imagem, o seu programa deve imprimir na tela o conteúdo de um arquivo PGM contendo o resultado da execução das duas operações acima, nesta sequência, sobre a imagem da entrada. Não é preciso imprimir a imagem resultante das operações em separado. O formato do arquivo PGM esperado é o mesmo do exemplo da entrada.

Observações

- É obrigatório utilizar uma lista encadeada para armazenar os elementos da matriz.
- Não é permitido armazenar elementos na lista encadeada cujo valor do pixel seja zero.
- Não é permitido fazer a inserção da borda nem a inversão da imagem durante o processo de impressão. Você deve inserir, excluir e modificar os elementos na lista. Somente no final deve ser feita a impressão, já com a lista contendo a imagem resultante.
- A ordem das operações deve ser a mesma da especificada no enunciado.
- É obrigatória a liberação de toda memória alocada dinamicamente que não for mais necessária, inclusive durante a execução do programa, quando os elementos forem removidos da lista encadeada.
- Você deve usar a seguinte linha de comando para executar o programa: **EP3 imagem-entrada.pgm imagem-saida.pgm**