MC202 - Estruturas de Dados
Semestre 2001-2 - Turmas EF
Prova 3: 04/dez/2001

RA	Nome
Assinatura	Notas

A prova é individual e sem consulta.
Não são permitidos computadores ou calculadoras.
Não separe as folhas deste caderno de prova.
Não é permitido o uso de outro rascunho além destas folhas.
Escreva seu nome completo, e assine a tinta.
Valem apenas as respostas nos espaços indicados.

1. [2.5 pontos] O *inverso* de um grafo G é o grafo G' que é igual a G exceto que todas as arestas têm o sentido invertido. Supondo a representação de grafos vista em classe, escreva um procedimento *inverte*(G, H) que coloca em H o inverso do grafo G, destruindo G no processo. O procedimento não deve usar nenhum vetor auxiliar, ou criar novos nós.

resposta

2. [2.5 pontos] Considere uma estrutura ligada onde cada registro, de tipo *Treco*, contém três apontadores *outro*, *amigo* e *compadre*, cada qual apontando para outro *Treco* (ou **nil**). Não há nenhuma restrição sobre esses apontadores, de modo que a estrutura pode ter ciclos, apontadores convergentes, etc.. Suponha que há também um campo inteiro *num* em cada nó, inicialmente -1. Escreva um procedimento que identifica todos os nós que podem ser atingidos a partir de um nó inicial $p\uparrow$ dado, seguindo-se qualquer combinação desses três apontadores, e atribui números consecutivos $1, 2, \ldots$ aos campos *num* desses nós, em ordem arbitrária.

resposta

3. [2.5 pontos] Escreva um procedimento para embaralhar os nós de uma lista duplamente ligada, em ordem aleatória. Suponha definida uma função $rand(m)$ que, a cada chamada, produz um novo inteiro aleatório entre 1 e m, inclusive. O procedimento não deve usar vetores auxiliares, ou alocar novos registros com **new**/**malloc**, e deve usar no máximo $O(n^2)$ operações para uma lista com n elementos.

resposta

4. [2.5 pontos] Escreva um procedimento que transforma uma árvore B numa árvore binária de busca. Suponha que cada nó da árvore B contém um contador $p\uparrow.m$, vetor de chaves $p\uparrow.\mathbf{ch}[0..p\uparrow.m-1]$, e um vetor de sub-árvores $p\uparrow.\mathbf{sub}[0..p\uparrow.m]$. A altura da árvore resultante deve ser $O(\log n)$, onde n é o número de chaves na árvore – mesmo que cada nó da árvore B contenha um número bem grande de chaves. *Dica*: use recursão, não iteração, para processar as chaves de um nó.

resposta