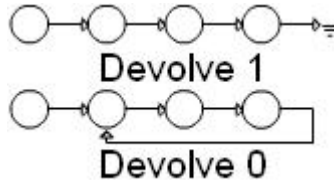


MAC 323 - Estruturas de Dados

Primeiro semestre de 2007

Quarta Lista de Exercícios – Entrega: 26 de junho de 2007

1. Crie uma função `INT LOOP(LISTANO* HEAD)` que recebe um ponteiro para a cabeça de uma lista simplesmente encadeada e devolva 1 caso a lista termine com `NULL`, ou 0 caso o último elemento da lista aponta para um nó interno. Você não sabe o tamanho da lista e ela pode conter valores repetidos em nós distintos.



2. Crie uma função `VOID INVERTEHEAP(INT *HEAP, INT N)` que recebe um ponteiro *heap* para um heap máximo (em que o máximo está no topo e cada elemento é maior que os filhos) de tamanho *n*, e transforme-o em um heap mínimo (em que o mínimo está no topo e cada elemento é menor que os filhos). Sua função deve ser tão eficiente quanto possível.
3. Crie uma função `ABBNO* MAC(ABBNO *PTR1, ABBNO *PTR2)` que receba dois ponteiros para dois nós de uma ABB e devolva o Menor Ancestral Comum dos dois. Considere que a função de comparação entre os valores da ABB é muito custosa, e que você deve evitá-la ao máximo. Considere também que um nó é ancestral dele mesmo.
4. Crie uma função `INT MENORDIFERENCA(INT **VALORES, INT *TAMANHOS, INT N)` que receba *n* vetores crescentes de inteiros, $valores[0], valores[1], \dots, valores[n-1]$, onde o vetor $valores[i]$ tem comprimento $tamanhos[i]$. Queremos formar um grupo com *n* elementos, um de cada vetor dado, de tal maneira que a diferença entre o maior e o menor valor do grupo seja mínima. Sua função deve devolver essa diferença. Suponha que todos os ponteiros sejam válidos e todos os vetores têm tamanho maior do que zero.
5. Crie uma função `INT MAISFREQUENTE(INT *VALORES, INT DIFERENTES, INT N)` que receba um vetor, muito grande, *valores* de *n* inteiros, mas que possui apenas *diferentes* valores distintos. Sua função deve devolver o valor que repete mais vezes.

Algumas notas finais:

- Defina as estruturas de dados que vai usar antes de fazer os exercícios.
- Considere que os nós de uma ABB têm somente os ponteiros DIREITA, ESQUERDA e PAI.
- Em momento algum será necessário fazer MALLOCs, ou tratamento de erro de entrada.
- Se precisar usar alguma outra estrutura de dados como pilha, fila, fila dupla, listas, ABBs, heaps, hash ou B-árvores, não precisa implementá-la.
- Coloque na resposta somente a função pedida, e as auxiliares que ela possa precisar. Não é necessário o código de todo o programa.
- Faça comentários em seu código para explicar o que não for óbvio. Exemplo: você não precisa explicar quando está alocando um novo nó. Mas é bom que explique quando está procurando por algo na lista.
- Evite entregar exercícios em branco. Se não conseguir resolver por completo, faça um *pseudo-rascunho-código* do que pensou.
- Tente resolver os exercícios da maneira mais eficiente possível. Todos têm soluções curtas e rápidas.
- Entregue sua lista preferencialmente em papel, em vez de usar a submissão online. Como sabem o Instituto está com alguns problemas na parte de impressão e, às vezes, é difícil conseguir imprimir as listas para corrigi-las.