

Cursos de Verão 2010 - IME/USP
Tópicos de Programação
3ª Lista de Exercícios - Data de Entrega: 12/02/2010
Prof. Alexandre da Silva Freire

Exercício C1 – Faça uma versão iterativa do algoritmo `pos_ordem` visto em aula.

Exercício C2 – Dado um vetor $v[0 \dots n - 1]$ de inteiros distintos, faça um algoritmo para inserir esses elementos em uma árvore binária de busca (inicialmente a árvore está vazia). Prove que a árvore resultante possui altura $O(\lg n)$.

Exercício C3 – Prove que o algoritmo `minimo` (encontra um elemento mínimo em uma árvore binária de busca), visto em aula, está correto.

Exercício C4 – Considere o seguinte algoritmo:

```
void em_ordem(ArvBin* arv) {
    em_ordem_rec(arv->raiz);
}

void em_ordem_rec(No* no) {
    if(no != NULL) {
        em_ordem_rec(no->esq);
        printf("%d\n", no->elem);
        em_ordem_rec(no->dir);
    }
}
```

Qual é o consumo de tempo do algoritmo `em_ordem`? Justifique sua resposta (defina uma recorrência, depois prove por indução qual é a “fórmula fechada” para a respectiva recorrência).

Exercício C5 – Neste exercício consideraremos um tipo de árvore binária que chamaremos de *árvore booleana*. Neste tipo de árvore, cada nível está completo, exceto possivelmente o último, e os nós contidos no último nível estão o mais à esquerda possível (esta é a definição de árvore binária quase completa, assim como vimos na definição de um heap). Além disso, cada nó da árvore terá ou 0 ou 2 filhos (nenhum nó tem apenas um filho).

O que torna especial uma árvore booleana é o fato de que cada nó é associado com um *valor* booleano (0 ou 1), e cada nó *interno* (que não é folha) é associado com um *operador* lógico (AND ou OR). O *valor* de um nó que está associado ao operador AND é dado pela operação AND entre os valores de seus dois filhos. Analogamente, o *valor* de um nó que está associado ao

operador OR é dado pela operação OR entre os valores de seus dois filhos. O *valor* de cada folha é 1 ou 0 (verdadeiro ou falso).

Escreva uma estrutura para representar tal árvore. Escreva um algoritmo que, dada uma árvore booleana na qual somente os valores das folhas estão definidos, preenche o campo *valor* de todos os nós.

Exercício C6 – Uma *árvore booleana mutável* é uma *árvore booleana* (definição dada no exercício anterior) na qual cada nó interno possui um campo `mutavel` que indica se podemos ou não trocar o operador do respectivo nó (de AND para OR ou vice-versa). Escreva uma estrutura para representar esta variação de árvore booleana. Escreva um algoritmo que, dada uma árvore booleana mutável na qual somente os valores das folhas estão definidos, encontre o número mínimo de nós cujos operadores precisam ser trocados para que a raiz da árvore tenha valor 1 (verdadeiro). Caso não seja possível fazer com que o valor da raiz seja 1, seu algoritmo deve devolver -1 .

Exercício C7 – Descreva um algoritmo que ordena um vetor $v[0 \dots n-1]$ em que todos os elementos pertencem a $\{0, 1, \dots, n^2 - 1\}$. O consumo de tempo do algoritmo deve ser $O(n)$. Justifique a sua resposta. Dica: considere a representação de cada elemento em $v[0 \dots n-1]$ na base n , que utiliza os algarismos $0, 1, \dots, n-1$. Ou seja, para $j = 0, 1, \dots, n-1$ temos que $v[j] = a_1n + a_0$.