

SEGUNDA PROVA DE ESTRUTURAS DE DADOS
BCC, 1o. SEMESTRE DE 2008

Instruções:

1. Não destaque as folhas do caderno de soluções.
2. A prova pode ser feita a lápis. Cuidado com a legibilidade.
3. Não é permitido o uso de folhas avulsas para rascunho.
4. Não é necessário apagar rascunhos no caderno de soluções.
5. Aserções imprecisas valem pouco. Justifique suas asserções (dentro do razoável!).

1. [4 pontos]

- (i) Suponha que inserimos, nesta ordem, as chaves `Q U E S T A O F C I L` em uma árvore 2-3-4 inicialmente vazia. Desenhe as árvores 2-3-4 que temos ao longo do processo. Desenhe pelo menos 11 árvores; para dizer como o árvore evolui, voce pode (deve?) desenhar árvores intermediárias/auxiliares.
- (ii) Suponha agora que inserimos, nesta ordem, as chaves `Q U E S T A O F C I L` em uma árvore rubro-negra esquerdista (ARNE) inicialmente vazia. Desenhe as ARNEs correspondentes às últimas 3 árvores 2-3-4 do item (i) (desenhe as 3 árvores resultantes da inserção de `C`, `I` e `L`).
- (iii) Descreva cuidadosamente como ocorre a inserção de `L` na sua ARNE (queremos saber aqui como a 2a. ARNE que você desenhou no item (ii) acima evolui para se tornar a 3a. ARNE que você desenhou). Para responder a este item, desenhe várias ARNEs (ou partes de ARNE) para ilustrar a execução do algoritmo de inserção em ARNEs. O código de inserção em ARNEs segue abaixo.

```
link RBinsert(link h, Item item)
{
    Key v = key(item);
    if (h == z)
        return NEW(item, z, z, 1); /* o 1 indica ponteiro vermelho */
    if (h->l->red && h->r->red) colorFlip(h);
    if (less(v, key(h->item))) h->l = insert(h->l, item);
        else h->r = insert(h->r, item);
    if (h->r->red) h = rotL(h);
    if (h->l->red && h->l->l->red) h = rotR(h);
    return h;
}

void STinsert(Item item)
{
    head = RBinsert(head, item); head->red = 0; }
}
```

2. [4 pontos]
- (i) Desenhe todas as possíveis árvores 2-3-4 para N itens, para $1 \leq N \leq 7$. (Use as chaves A B C D E F G.) [*Observação.* Para $N = 3$, temos duas tais árvores e para $N = 7$ temos 9.]
 - (ii) Suponha que inserimos as chaves A B C D E em uma das $5! = 120$ ordenações possíveis, com todas as possibilidades equiprováveis, em uma árvore 2-3-4 inicialmente vazia. Diga quais são as possíveis árvores que podemos obter nesse processo.
 - (iii) Suponha que inserimos as chaves A B C D E em uma das $5! = 120$ ordenações possíveis, com todas as possibilidades equiprováveis, em uma ARNE inicialmente vazia. Quais ARNEs com 5 nós internos podemos obter? Diga a probabilidade de obtermos cada uma dessas árvores.
3. [2 pontos] Suponha que inserimos, nesta ordem, as chaves Q U E S T A O F C I L em uma tabela de hashing inicialmente vazia com $M = 12$ entradas. Suponha que usamos a função de hashing $13k \bmod M$ para transformar a k -ésima letra do alfabeto no endereço da tabela (por exemplo, A é levado ao endereço 1 e S é levado ao endereço 7). Mostre claramente a evolução da tabela nesse processo. Considere os seguintes dois casos:
- (i) Neste item, suponha que usamos resolução de colisões por encadeamento (*separate chaining*).
 - (ii) Neste segundo item, suponha que usamos resolução de colisões por *linear probing*.
4. [1 pontos] Desenhe a *trie* que obtemos ao inserirmos as seguintes chaves em uma *trie* inicialmente vazia:

```

01111100
10101100
00011101
01011101
10111100
01111101
01101100
11011100

```

Nesta questão, você não precisa desenhar cada etapa do processo de inserção; basta desenhar o resultado final.