PROVA SUBSTITUTIVA DE ESTRUTURAS DE DADOS
BCC, 1o. SEMESTRE DE 2012

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. Asserções imprecisas valem pouco. Justifique suas asserções (dentro do razoável).

Importante: faça no máximo três questões.

1. [3 pontos] Um dos objetivos principais desta disciplina é estudar a noção de tabela de símbolos e investigar várias implementações possíveis.
   (*i*) Descreva sucintamente uma aplicação de tabelas de símbolos. Seu exemplo deve ser interessante do ponto de vista prático ou do ponto de vista teórico. Seu exemplo deve ser substancialmente diferente dos EPs dados durante o semestre.
   (*ii*) No EP4, você comparou várias implementações de tabelas de símbolos experimentalmente e escreveu um relatório com seus resultados experimentais e suas conclusões. Reproduza aqui os pontos principais de seu relatório. (Não deixe de descrever em detalhe quais experimentos você executou para comparar as várias tabelas de símbolos que você implementou.)
2. [3 pontos]
   (*i*) Defina precisamente o Problema da Cobertura Exata (PCE) e o Problema da Cobertura Generalizado (PCG). Dê um exemplo concreto, ilustrativo mas pequeno do PCG. Dê todas as soluções possíveis de seu exemplo.
   (*ii*) O problema das N rainhas Q_N consiste em colocar N rainhas em um tabuleiro de xadrez $N \times N$, sem que nenhuma rainha ataque outra.[1] Exiba todas as soluções desse problema para $N = 1, \ldots, 4$.
   (*iii*) Descreva precisamente uma redução de Q_N para o PCG para um N genérico. Ilustre sua redução para $N = 3$.
3. [2 pontos]
   (*i*) Suponha que inserimos, nesta ordem, as chaves `U M A Q E T O P D R` em uma skip list inicialmente vazia. Suponha que o gerador de números aleatórios fornece os seguintes resultados, nessa ordem: `1 4 5 2 3 3 6 1 2 3` (assim, p.ex., a célula de `A` fica com 5 ponteiros). Desenhe o resultado final dessas 10 inserções.
   (*ii*) Descreva detalhadamente como funciona o processo de inserção da chave `S` na skip list que você obteve em (*i*); suponha que o gerador de números aleatórios especificou que

Date: Versão de 3 de julho de 2012.

[1] Uma rainha ataca outra se elas estão na mesma linha, na mesma coluna, ou na mesma diagonal.

esta chave deverá ser inserida em uma célula com 5 ponteiros. Em particular, diga quais são todas as comparações entre chaves que são executadas nesta inserção.

4. [4 pontos] Esta questão trata de árvores rubro-negra esquerdistas (ARNEs), usadas para implementar tabelas de símbolos. Por simplicidade, suponha que os itens e chaves são inteiros.
   (*i*) Suponha que inserimos, nesta ordem, as chaves `11 44 99 55 22 88 33 66 77` em uma ARNE inicialmente vazia. Desenhe as 9 ARNEs que resultam das 9 inserções acima. Não desenhe árvores 2–3; desenhe ARNEs e execute as inserções seguindo rigorosamente a rotina de inserção em ARNEs.
   (*ii*) Escreva uma função de protótipo `void STprint_range(Key lo, Key hi)` que imprime, em ordem crescente, todos os itens `x` na ARNE com `lo` $\leq$ `x` $\leq$ `hi`. Para tanto, escreva e use uma função recursiva de protótipo `void print_rangeR(link h, Key lo, Key hi)`. Sua função não deve percorrer "partes desnecessárias" da ARNE. O que ocorre quando executamos `STprint_range(lo, hi)` com `lo` e `hi` iguais?
   (*iii*) Qual é a complexidade de tempo da função `STprint_range()` que você implementou? Sua resposta deve ser em função do número total de elementos N na ARNE e do número de elementos entre `lo` e `hi` na ARNE. Justifique sua resposta.

As duas rotinas principais envolvidas na inserção em ARNEs são dadas a seguir.

```
link LLRBinsert(link h, Item item)
  { Key v = key(item);

   /* Insert a new node at the bottom*/
    if (h == z) return NEW(item, z, z, 1, 1);

    if (less(v, key(h->item))) hl = LLRBinsert(hl, item);
                          else hr = LLRBinsert(hr, item);

    /* Enforce left-leaning condition */
    if (hr->red && !hl->red) h = rotL(h);
    if (hl->red && hll->red) h = rotR(h);
    if (hl->red && hr->red) colorFlip(h);

    return fixNr(h);
  }

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