

Lista de Exercícios — Árvores Rubro-Negras

QXD0115 – Estrutura de Dados Avançada – 2024

Prof. Atílio Gomes

6 de julho de 2024

Aluno: [] Matrícula: []

Referências: Cormem et al. Introduction to Algorithms

1. Seja `INSERT(int chave)` a função de inclusão de nós em uma árvore binária de busca simples. Desenhe a árvore que resulta após a chamada a `INSERT` na árvore rubro-negra da Figura 1 abaixo com chave 36. Se o nó inserido for vermelho, a árvore resultante é uma árvore rubro-negra? E se ele for preto?

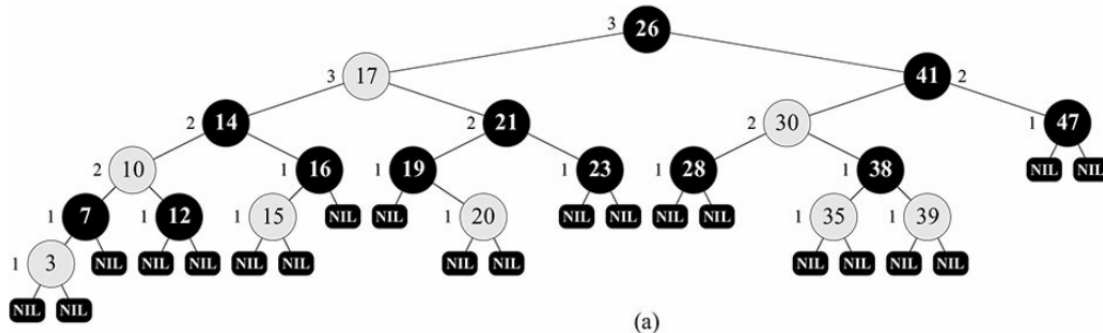


Figura 1: Uma árvore rubro-negra com nós pretos em preto e nós vermelhos em cinzento. Todo nó em uma árvore rubro-negra é vermelho ou preto, os filhos de um nó vermelho são pretos, e todo caminho simples de um nó até uma folha descendente contém o mesmo número de nós pretos. Toda folha, mostrada como um NIL, é preta. Cada nó não NIL é marcado na figura com sua altura negra: nós NILs têm altura negra igual a 0.

2. Vamos definir uma árvore rubro-negra **relaxada** como uma árvore de busca binária que satisfaz as propriedades rubro-negras 1, 3, 4 e 5. Em outras palavras, a raiz pode ser vermelha ou preta. Considere uma árvore rubro-negra relaxada T cuja raiz é vermelha. Se colorirmos a raiz de T de preto, mas não fizermos nenhuma outra mudança em T , a árvore resultante é uma árvore rubro-negra?
3. Desenhe uma árvore rubro-negra que contém dois caminhos da raiz até as folhas tal que um caminho seja o dobro do tamanho do outro.
4. Mostre que o comprimento do caminho mais longo de um nó x em uma árvore rubro-negra até uma de suas folhas descendentes é, no máximo, duas vezes o do caminho mais curto do nó x até uma folha descendente.
5. Qual é o maior número possível de nós internos em uma árvore rubro-negra com altura negra k ? Qual é o menor número possível?
6. Prove ou dê um contra-exemplo.
 - (a) toda árvore Rubro-Negra é uma árvore AVL.
 - (b) toda árvore AVL é uma árvore Rubro-Negra.
7. Suponha que x é um nó de uma árvore binária de busca rubro-negra. Suponha que `x->left == nullptr` mas `x->esq != nullptr`. Prove que `x->left->color == RED` e `x->left` não tem filhos (ou seja, `x->left->left == nullptr` e `x->left->right == nullptr`).
8. Escreva um código que calcula a altura (total) e a altura negra de uma árvore binária de busca rubro-negra.
9. Dada uma árvore rubro-negra T arbitrária e um nó x qualquer de T , escreva o pseudocódigo para a rotação direita da subárvore com raiz x .
10. Demonstre que, em toda árvore de busca binária de n nós, existem exatamente $n - 1$ rotações possíveis.
11. Sejam a , b e c nós arbitrários nas subárvores α , β e γ , respectivamente, na árvore da direita da Figura 2. Como as profundidades de a , b e c mudam quando é realizada uma rotação para a esquerda no nó x na figura?

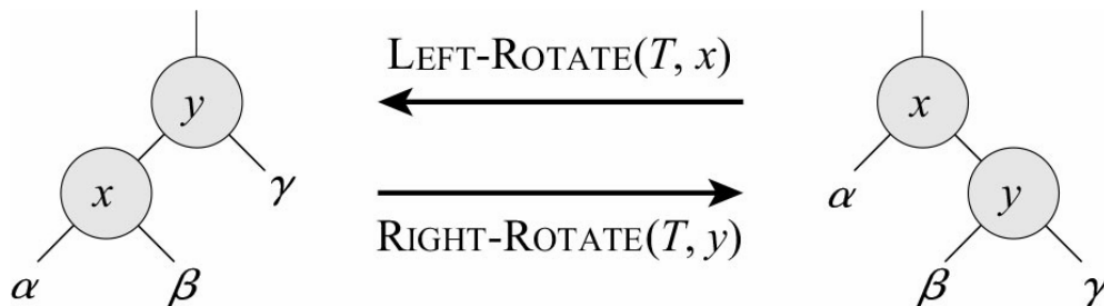


Figura 2: As operações de rotação em uma árvore de busca binária.

12. Mostre que qualquer árvore de busca binária arbitrária T com n nós pode ser transformada em qualquer outra árvore de busca binária arbitrária T' com n nós por meio de $O(n)$ rotações. (Sugestão: Primeiro, mostre que, no máximo, $n - 1$ rotações para a direita são suficientes para transformar a árvore T em uma outra árvore T'' em que a seguinte propriedade é verdadeira para todo nó v de T'' : o filho esquerdo de v é vazio e se v tem um filho não-vazio então esse filho é o filho direito de v .)
13. Note que no algoritmo de inclusão de um novo nó z na árvore rubro-negra, atribuímos a cor VERMELHO ao nó z recém-inserido. Note que, se tivéssemos optado por atribuir a cor preta a z , a propriedade 4 de uma árvore rubro-negra não seria violada. Por que não optamos por definir z como preto?
14. Mostre as árvores rubro-negras que resultam após a inserção sucessiva das chaves 41, 38, 31, 12, 19, 8 em uma árvore rubro-negra inicialmente vazia.
15. Insira em uma árvore Rubro-Negra, itens com as chaves: 4 – 7 – 12 – 15 – 3 – 5 – 14 – 18 (nesta ordem). Desenhe a árvore resultante da inserção, sendo que uma nova árvore deve ser desenhada quando houver uma rotação ou troca de cores. (Atenção: verifique a necessidade de rotação e/ou troca de cores a cada inserção)
16. Desenhe a sequência de árvores binárias de busca rubronegras que resulta da inserção das chaves E A S Y Q U T I O N, nesta ordem, em uma árvore inicialmente vazia. (Desenhe apenas a árvore binária de busca no final de cada inserção; não desenhe as árvores binárias de busca temporárias que aparecem durante a operação.)
17. O aluno Roberto está preocupado que a operação de inclusão de nós RN-INSERT-FIXUP, vista em sala, possa atribuir a cor VERMELHO ao nó $T.nil$, caso em que o teste da linha 1 do algoritmo não faria o laço terminar quando z fosse a raiz. Mostre que a preocupação do aluno é infundada, demonstrando que RN-INSERT-FIXUP nunca atribui cor VERMELHO a $T.nil$.

18. As árvores lembram de sua história. Mostre que árvores binárias de busca rubro-negras lembram de sua história. Por exemplo, se você inserir uma chave que é menor que todas as chaves da árvore e logo em seguida remover o mínimo da árvore, você pode obter uma árvore diferente da original.