

Universidade Federal do Pará
Instituto de Ciências Exatas e Naturais
Programa de Pós-Graduação em Ciência da Computação
Projeto e Análise de Algoritmos

Lista de Exercícios

Árvores Balanceadas, TRIE & PATRICIA e Tabelas de Dispersão

01) [POSCOMP 2010] Assinale a alternativa em que todas as propriedades de uma árvore vermelho e preto são verdadeiras.

(a) Todo nó é vermelho ou preto. A raiz pode ser vermelha ou preta. Todas as folhas são vermelhas.

(b) A raiz é preta. Todas as folhas são vermelhas. Para cada nó, todos os caminhos, desde um nó até as folhas descendentes, contêm um mesmo número de nós pretos.

(c) Toda folha é preta. Todo nó é vermelho ou preto. A raiz é preta.

(d) Se um nó é vermelho, ambos os filhos são vermelhos. A raiz pode ser vermelha ou preta. Todas as folhas são pretas.

(e) Todas as folhas são vermelhas. Todo nó é vermelho ou preto. A raiz pode ser vermelha ou preta.

02) [POSCOMP 2005] Dadas as seguintes características para uma Árvore B de ordem n :

(I) Toda página contém no máximo $2n$ itens (chaves).

(II) Toda página, exceto a página raiz, contém no mínimo n itens.

(III) Toda página ou é uma página folha, ou tem $m + 1$ descendentes, onde m é o número de chaves.

(IV) Todas as páginas folhas aparecem no mesmo nível.

Qual das seguintes opções é verdadeira?

(a) As características (I), (II), (III) e (IV) são falsas.

(b) Apenas as características (I) e (IV) são verdadeiras.

(c) Apenas as características (II), (III) e (IV) são verdadeiras.

(d) As características (I), (II), (III) e (IV) são verdadeiras.

(e) As características (II), (III) e (IV) são falsas.

03) [POSCOMP 2007] Seja T uma árvore AVL vazia. Supondo que os elementos 5, 10, 11, 7, 9, 3 e 6 sejam inseridos nessa ordem em T , indique a sequência abaixo que corresponde a um percurso de T em pós-ordem.

(a) 3, 5, 6, 7, 9, 10 e 11.

(b) 7, 5, 3, 6, 10, 9 e 11.

(c) 9, 10, 7, 6, 11, 5 e 3.

(d) 11, 10, 9, 7, 6, 5 e 3.

(e) 3, 6, 5, 9, 11, 10 e 7.

04) [POSCOMP 2010] Em uma Árvore B de ordem m , temos que: (i) cada nó contém no mínimo m registros (e $m + 1$ descendentes) e no máximo $2m$ registros (e $2m + 1$ descendentes), exceto o nó raiz que pode conter entre 1 e $2m$ registros; (ii) todas os nós folha aparecem no mesmo nível. Sobre Árvores B, é correto afirmar que

- (a) o particionamento de nós ocorre quando um registro precisa ser inserido em um nó com $2m$ registros.
- (b) o particionamento de nós ocorre quando um registro precisa ser inserido em um nó com menos de $2m$ registros.
- (c) o particionamento de nós ocorre quando a chave do registro a ser inserido contém um valor (conteúdo) intermediário entre os valores das chaves dos registros contidos no mesmo nó.
- (d) o particionamento de nós ocorre quando é necessário diminuir a altura da árvore.
- (e) a sua altura aumenta em um nível toda vez que ocorre o particionamento de um nó.

05) [POSCOMP 2009] Quais das seguintes propriedades **não** se aplicam a árvores rubro-negras?

- (a) Todo nó é vermelho ou preto.
- (b) Todo nó folha é preto.
- (c) Se um nó é preto, ambos seus filhos são vermelhos.
- (d) Se um nó é vermelho, ambos seus filhos são negros.
- (e) Todos os caminhos simples entre um nó e suas folhas descendentes contêm o mesmo número de nós pretos.

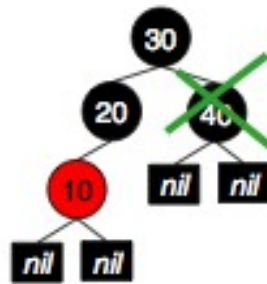
06) [POSCOMP 2011] Ao usar o cálculo de endereço ou *hashing*, geralmente é necessário o uso de um método de tratamento de colisões. Sobre esse método, é correto afirmar que

- (a) o tratamento de colisões é necessário apenas quando a tabela está cheia e se necessita inserir mais uma chave.
- (b) o tratamento de colisões é necessário para determinar o local da chave no momento da inserção na tabela.
- (c) o tratamento de colisões é necessário quando a tabela está vazia, pois não é possível calcular o endereço diretamente nesse caso.
- (d) o tratamento de colisões é necessário quando a chave inserida ainda não existir na tabela de endereçamento.
- (e) o tratamento de colisões é necessário, pois o *hashing* gera repetição de endereço para diferentes chaves.

07) [POSCOMP 2002] Considere uma tabela de espalhamento (tabela de *hash*) com quatro posições numeradas 0, 1, 2 e 3. Se a sequência de quadrados perfeitos 1, 4, 9, ..., i^2 , ... for armazenada nessa tabela segundo a função $f(x) = x \bmod 4$, como se dará a distribuição dos elementos pelas posições da tabela à medida que o número de entradas cresce?

- (a) Cada posição da tabela receberá aproximadamente o mesmo número de elementos.
- (b) Três posições da tabela receberão, cada uma, aproximadamente um terço dos elementos.
- (c) Uma única posição da tabela receberá todos os elementos, e as demais posições permanecerão vazias.
- (d) Todas as posições da tabela receberão elementos, mas as duas primeiras receberão, cada uma, o dobro das outras.
- (e) As duas primeiras posições da tabela receberão, cada uma, aproximadamente a metade dos elementos, e as demais posições permanecerão vazias.

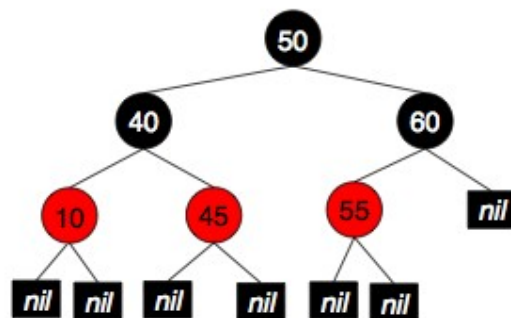
08) Dada a árvore vermelho-preto abaixo, remova o elemento com chave 40 e mantenha seu balanceamento.



A árvore vermelho-preto resultante em pós-ordem é

- (a) 20(P), 10(V), 30(P).
- (b) 10(P), 30(P), 20(P).
- (c) 20(P), 10(P), 30(P).
- (d) 10(P), 20(P), 30(P).
- (e) 20(V), 10(P), 30(P).

09) Dada a árvore vermelho-preto abaixo, remova o elemento com chave 50 e mantenha seu balanceamento.



A árvore vermelho-preto resultante em pré-ordem é

- (a) 60(P), 40(P), 10(V), 45(V), 55(P).
- (b) 10(V), 40(P), 45(V), 55(P), 60(P).
- (c) 55(P), 40(P), 10(V), 45(V), 60(P).
- (d) 55(P), 40(V), 10(P), 45(P), 60(V).
- (e) 10(V), 45(V), 40(P), 60(P), 55(P).

10) [POSCOMP 2006] Sejam [6, 4, 2, 1, 3, 5, 8, 7, 9] e [7, 4, 3, 2, 1, 6, 5, 10, 9, 8, 11] as sequências produzidas pelo percurso em pré-ordem das árvores binárias de busca T1 e T2, respectivamente. Marque a afirmação **incorreta**.

- (a) T1 possui altura mínima dentre todas as árvores binárias com 9 nós.
- (b) T1 é uma árvore AVL.
- (c) T1 é uma árvore rubro-negra.
- (d) T2 possui altura mínima dentre todas as árvores binárias com 11 nós.
- (e) T2 não é uma árvore AVL.

11) Analise as afirmativas abaixo.

I. Organizações de arquivos baseadas em *hashing* permitem-nos encontrar o endereço de um item de dado diretamente por meio do cálculo de uma função sobre o valor da chave de procura do registro desejado.

II. A árvore Patricia é obtida a partir de uma árvore digital binária de prefixo, onde nenhuma chave é prefixo de outra chave, e a sequência de nós com apenas um filho são compactados em um único nó.

III. A técnica da árvore digital é tão mais eficiente quanto maior for a quantidade de chaves com prefixos comuns.

A análise permite concluir que

- (a) todas as afirmativas são falsas.
- (b) todas as afirmativas são verdadeiras.
- (c) apenas as afirmativas (I) e (II) são verdadeiras.
- (d) apenas as afirmativas (I) e (III) são verdadeiras.
- (e) apenas as afirmativas (II) e (III) são verdadeiras.

12) [POSCOMP 2009] Considere uma tabela de espalhamento (tabela *hash*) de comprimento $m = 11$, que usa endereçamento aberto (*open addressing*), a técnica de tentativa linear (*linear probing*) para resolver colisões e com a função de dispersão (*função hash*) $h(k) = k \bmod m$, onde k é a chave a ser inserida. Considere as seguintes operações sobre essa tabela:

- Inserção das chaves 3, 14, 15, 92, 65, 35 (nesta ordem);
- Remoção da chave 15; e
- Inserção da chave 43.

Escolha a opção que representa esta tabela após estas operações:

- (a) 65 – \emptyset – 35 – 14 – \emptyset – 92 – 3 – \emptyset – \emptyset – \emptyset – 43.
- (b) 43 – \emptyset – 35 – 3 – 14 – 92 – \emptyset – \emptyset – \emptyset – \emptyset – 65.
- (c) 65 – \emptyset – 35 – X – 14 – 92 – 3 – \emptyset – \emptyset – \emptyset – 43.
- (d) 65 – \emptyset – 35 – 3 – 14 – 92 – \emptyset – \emptyset – \emptyset – \emptyset – 43.
- (e) 43 – \emptyset – 35 – 3 – 14 – X – 92 – \emptyset – \emptyset – \emptyset – 65.

13) Seja T uma árvore rubro-negra vazia. Agora, suponha que os elementos 7, 11, 2, 1, 6, 8, 14, 4 e 5 sejam inseridos nessa ordem em T e, no fim, o elemento 5 seja removido após ter sido inserido por engano. Indique a sequência abaixo que corresponde a um percurso de T em pré-ordem.

- (a) 1(P), 4(V), 6(P), 2(V), 8(V), 14(V), 11(P) e 7(P).
- (b) 1(P), 2(V), 4(V), 6(P), 7(P), 8(V), 11(P) e 14(V).
- (c) 7(P), 2(V), 1(P), 6(P), 4(V), 11(P), 8(V) e 14(V).
- (d) 7(P), 2(V), 1(P), 4(P), 6(P), 11(P), 8(V) e 14(V).
- (e) 7(P), 2(P), 1(P), 6(V), 4(P), 11(P), 8(V) e 14(V).

14) Analise as afirmativas abaixo.

I. Uma das vantagens da pesquisa baseada em *hashing* é a sua eficiência, dado que a ordem do pior caso e do caso esperado de tempo de execução é $O(n)$ e $O(1)$, respectivamente.

II. A tabela *hash* deve ser utilizada quando o objetivo é ter eficiência nas operações de pesquisa, inserção e remoção, desde que o número de inserções e remoções não provoque variações grandes no número de registros da tabela.

III. A tabela *hash* é indicada quando não há a necessidade de considerar a ordem das chaves e de saber a posição da chave de pesquisa em relação a outras chaves.

A análise permite concluir que

- (a) todas as afirmativas são verdadeiras.
- (b) apenas as afirmativas (I) e (II) são verdadeiras.
- (c) apenas as afirmativas (I) e (III) são verdadeiras.
- (d) apenas as afirmativas (II) e (III) são verdadeiras.
- (e) todas as afirmativas são falsas.

15) Obtenha a árvore PATRICIA a partir da árvore TRIE binária abaixo de altura 6.

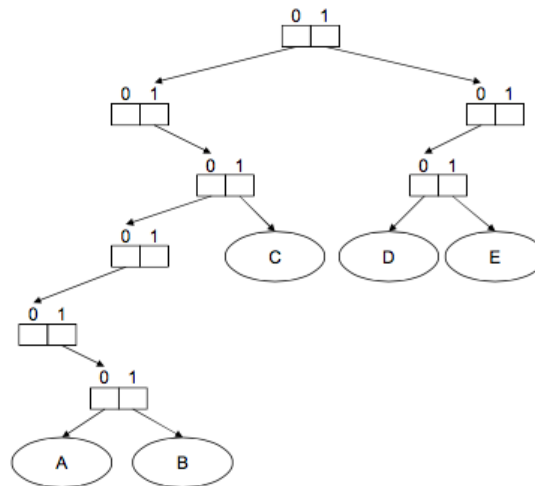
■ A = 010010

■ B = 010011

■ C = 011000

■ D = 100001

■ E = 101000



Sobre a árvore PATRICIA resultante, é correto afirmar que

- (a) a sua altura é igual a 4.
- (b) a sua altura é igual a 4 após a inserção da chave F = 111111.
- (c) a sua altura aumenta de uma unidade após a inserção da chave F = 111111.
- (d) não é possível a inserção da chave F = 1101.
- (e) a sua altura é igual a 3 após a inserção da chave F = 1101.