Questões Verdadeiro ou Falso - Árvores (30 questões)

Instruções

Marque as afirmações abaixo como Verdadeiras (V) ou Falsas (F).

Questão 1: A codificação de Huffman é lossless (sem perdas). **( ) Verdadeiro ( ) Falso**

Questão 2: Na codificação de Huffman, um código nunca é prefixo de outro. **( ) Verdadeiro ( ) Falso**

Questão 3: Tries são eficazes para operações de busca de prefix em strings. **( ) Verdadeiro ( ) Falso**

Questão 4: Tabelas hash ocupam menos espaço de memória do que árvores binárias de pesquisa. **( ) Verdadeiro ( ) Falso**

Questão 5: Fazer uma busca em uma árvore binária de pesquisa pode ser tão lento quanto em uma lista encadeada. **( ) Verdadeiro ( ) Falso**

Questão 6: Em uma busca em profundidade do tipo 'pós-ordem' em árvore, o último elemento a ser visitado (supondo que a busca imprima os elementos ao visitá-los) será o maior elemento da árvore. **( ) Verdadeiro ( ) Falso**

Questão 7: Achar o menor elemento de uma tabela hash (implementada de forma tradicional) é mais rápido do que achar o menor elemento em uma árvore binária de pesquisa. **( ) Verdadeiro ( ) Falso**

Questão 8: Todos os nós de uma árvore B onde cada nó possui capacidade para 4 elementos devem possuir pelo menos 2 elementos. **( ) Verdadeiro ( ) Falso**

Questão 9: Uma árvore AVL sempre tem altura menor ou igual a $1.44 \times \log_2(n+2)$, onde n é o número de nós. **( ) Verdadeiro ( ) Falso**

Questão 10: Em uma árvore Red-Black, o caminho mais longo da raiz até uma folha tem no máximo o dobro do comprimento do caminho mais curto. **( ) Verdadeiro ( ) Falso**

Questão 11: A operação de inserção em uma árvore AVL pode requerer no máximo uma rotação para rebalancear a árvore. **( ) Verdadeiro ( ) Falso**

Questão 12: Uma árvore binária de pesquisa degenerada (em forma de lista) tem complexidade O(n) para busca, inserção e remoção. **( ) Verdadeiro ( ) Falso**

Questão 13: Em uma árvore binária completa com n nós, a altura é sempre $\lfloor \log_2 n \rfloor$. **( ) Verdadeiro ( ) Falso**

Questão 14: A remoção de um nó em uma std::map nunca invalida iteradores que apontam para outros elementos. **( ) Verdadeiro ( ) Falso**

Questão 15: Em uma árvore trie, o número máximo de nós é limitado pelo comprimento da string mais longa inserida. **( ) Verdadeiro ( ) Falso**

Questão 16: Uma árvore splay garante que qualquer sequência de m operações em uma árvore com n nós executa em O(m log n) tempo amortizado. **( ) Verdadeiro ( ) Falso**

Questão 17: O percurso em-ordem de uma árvore binária de pesquisa sempre produz os elementos em ordem crescente. **( ) Verdadeiro ( ) Falso**

Questão 18: Em uma árvore de segmentos, é possível executar range queries e range updates em O(log n) tempo. **( ) Verdadeiro ( ) Falso**

Questão 19: Uma árvore B+ sempre armazena todos os dados nas folhas, enquanto nós internos contêm apenas chaves de indexação. **( ) Verdadeiro ( ) Falso**

Questão 20: A operação ++iterator em uma std::set sempre executa em O(1) tempo amortizado. **( ) Verdadeiro ( ) Falso**

Questão 21: Uma árvore binária pode ser reconstruída unicamente a partir apenas de seu percurso pré-ordem. **( ) Verdadeiro ( ) Falso**

Questão 22: Em uma árvore Fenwick (Binary Indexed Tree), todas as operações de update e query executam em O(log n) tempo. **( ) Verdadeiro ( ) Falso**

Questão 23: Uma árvore cartesiana construída a partir de um array sempre mantém a propriedade de heap para as prioridades. **( ) Verdadeiro ( ) Falso**

Questão 24: O fator de balanceamento de qualquer nó em uma árvore AVL está sempre no intervalo [-1, 0, +1]. **( ) Verdadeiro ( ) Falso**

Questão 25: Em uma árvore Red-Black, todos os caminhos da raiz até as folhas têm o mesmo número de nós pretos. **( ) Verdadeiro ( ) Falso**

Questão 26: A operação de merge de duas árvores AVL pode ser implementada em O(log n) tempo no pior caso. **( ) Verdadeiro ( ) Falso**

Questão 27: Uma árvore de sufixos para uma string de tamanho n sempre contém exatamente n sufixos. **( ) Verdadeiro ( ) Falso**

Questão 28: Em uma treap, a estrutura da árvore é determinada apenas pelas prioridades dos nós, não pelos valores das chaves. **( ) Verdadeiro ( ) Falso**

Questão 29: A operação erase(iterator) em uma std::map sempre retorna um iterador válido para o próximo elemento. **( ) Verdadeiro ( ) Falso**

Questão 30: Uma árvore de intervalos pode encontrar todos os intervalos que se sobrepõem com um dado intervalo em O(k + log n) tempo, onde k é o número de intervalos encontrados. **( ) Verdadeiro ( )**

Falso