

Departamento de Informática - UFPR
Algoritmos e Estruturas de Dados III - CI057 - 2021/2 (2022)
Segunda prova
Prof. Eduardo Almeida

-
- A prova tem duração de 1h30m.
 - A interpretação faz parte da prova. Pode fazer a lápis.
 - PROVA SEM CONSULTA.
-

(40pts) 1. Marque (V) para verdadeiro e (F) para falso nas questões à seguir. Caso marque (F) justifique a resposta. Use a letra correspondente na folha de respostas.

- i (V) Uma árvore B+ aceita chaves duplicadas.
- ii (V) Dada a função hash $h(k, i) = (10 * k + i) \bmod 11$: a inclusão das chaves 21, 28, 15, 20, 33, 12, 19 e 10 em uma tabela hash de endereçamento aberto com $m = 11$ posições gera agrupamento primário.
- iii (F) O texto “avante elefante”, sem as aspas, será comprimido em 40 bits utilizando o algoritmo de Huffman apresentado em sala.
- iv (V) Considerando uma árvore Trie 26-ária, a ordem inclusão de um conjunto de palavras não altera o formato da árvore final.
- v (F) Considerando todos os elementos distintos em uma heap mínima, o menor elemento do arranjo pode residir em qualquer uma das folhas.
- vi () Uma busca por intervalo de chaves em uma tabela hash é muito mais eficiente que a mesma busca em uma árvore-B+ (Por ex. buscar todos os nós com as chaves entre 10 e 20).
- vii (V) Dada uma árvore-B+ com $t = 3$, a raiz somente armazena chaves ímpares após inclusão das chaves íteiras de 1 até 10.
- viii (V) Uma tabela hash aceita chaves duplicadas.

(20pts) 2. Em uma tabela hash com 100 entradas, as colisões são resolvidas usando listas encadeadas. Para reduzir o tempo de busca, decidiu-se que cada lista seria reorganizada como uma árvore binária de busca. A função utilizada é $h(k) = k \bmod 100$. Infelizmente, as chaves inseridas seguem o padrão $k_i = 50i$, onde k_i , corresponde à i -ésima chave inserida. Mostre a situação da tabela após a inserção de k_i , com $i = 1, 2, 3, \dots, 13$. Faça o desenho da tabela resultante.

(40pts) 3. Considere a estrutura abaixo para representar uma árvore TRIE N-ária. As chaves armazenadas na árvore são compostas pelas letras a-z. Cada nó da estrutura possui 26 apontadores para os filhos, sendo que a posição 0 do vetor de apontadores representa a letra 'a', a posição 1 representa a letra 'b', e assim por diante. A função $\text{ord}(x)$ retorna o valor do índice do vetor que corresponde ao caractere x. Por exemplo: $\text{ord}('a')=0$, $\text{ord}('b')=1$, $\text{ord}('x')=24$. Escreva um algoritmo em C ou em pseudocódigo semelhante a C que recebe como entrada uma árvore TRIE que utiliza a estrutura abaixo e uma chave, e retorna a chave sucessora da chave passada como parâmetro. A chave sucessora é dada pela próxima chave na ordenação lexicográfica das chaves armazenadas na árvore. Por exemplo, se as seguintes chaves estão armazenadas na árvore: c-a-m-a-d-a , c-a-n-a , c-a-m-a , o sucessor de c-a-m-a é c-a-m-a-d-a .

```
typedef struct tNo *ApontadorNo;
typedef struct tNo {
    char *valor;
    ApontadorNo ponteiros[26];
} tNo;
```
