

Instruções para a prova

- A prova é sem consulta;
- A prova dura 1 hora e 40 minutos.

***Questão 1 (25 pontos).** Comprima o texto “abracadabra”, sem as aspas, utilizando o algoritmo de Huffman apresentado em sala. Você deve apresentar a árvore binária gerada e a quantidade de bits originais e comprimidos.

***Questão 2 (25 pontos)** 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.

a -(F) - O nodo interno de nível mais baixo (1 acima da folha) de uma árvore B+ armazena chaves e valores.

b -(F) - A sequência {27, 17, 14, 6, 13, 10, 1, 5, 7, 12} é um heap máximo

c -(V) - Em qualquer subárvore de um heap mínimo, a raiz da subárvore contém valor menor que ocorre em qualquer lugar nessa subárvore.

d -(V) - Considerando todos os elementos distintos em uma heap máxima, o menor elemento do arranjo pode residir em qualquer uma das folhas.

e -(F) - Dada a função hash $h(k) = k \% m$: a inclusão das chaves 5, 28, 15, 20, 34, 12, 8, 19 e 10 em uma tabela hash com $m = 9$ posições não gera colisões.

7. **Questão 3 : (20 pontos)**

O TTL (time-to-leave) é usado como mecanismo para descartar dados que circulam em uma rede após um certo tempo, para evitar que o dado circule indefinidamente. Dados com os menores tempos são descartados.

Desenvolva um algoritmo em C (ou pseudo código similar), baseado em Heap, que leia e extraia o valor do menor TTL (simulando o descarte). O pseudo-código de ajuste de Heap máximo abaixo pode ser usado como referência.

```
ajustaHeap (A, i) {
    e = esq(i); d = dir(i);
    if e <= tamHeap(A) and A[e] > A[i]
        maior = e
    else
        maior = i
    if d <= tamHeap(A) and A[d] > A[maior]
        maior = d
    if maior <> i
        troca(A[i], A[maior])
        heapfy-max (A, maior)
}
```

Questão 4 : (30 pontos)

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 $ord(x)$ retorna o valor do índice do vetor que corresponde ao caracter x. Por exemplo: $ord('a')=0$, $ord('b')=1$, $ord('x')=24$.

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

Escreva um algoritmo em C ou em pseudocódigo semelhante a C que recebe como entrada uma árvore TRIE que utiliza a estrutura acima e uma chave, e que retorna o tamanho da menor chave da árvore.

