
**PROCESSO SELETIVO
VAGAS RESIDUAIS 2016
UFBA**

06

ESTRUTURA DE DADOS

**CIRCUITOS DIGITAIS E ARQUITETURA
DE COMPUTADORES**

REDAÇÃO

INSTRUÇÕES

Para a realização das provas, você recebeu este Caderno de Questões, uma Folha de Respostas para as Provas I e II e uma Folha de Resposta destinada à Redação.

1. Caderno de Questões

- Verifique se este Caderno de Questões contém as seguintes provas:
Prova I: ESTRUTURA DE DADOS — Questões de 01 a 35
Prova II: CIRCUITOS DIGITAIS E ARQUITETURA DE COMPUTADORES — Questões de 36 a 70
Prova de REDAÇÃO
- Qualquer irregularidade constatada neste Caderno de Questões deve ser imediatamente comunicada ao fiscal de sala.
- Nas Provas I e II, você encontra apenas um tipo de questão: objetiva de proposição simples. Identifique a resposta correta, marcando na coluna correspondente da Folha de Respostas:

V, se a proposição é verdadeira;

F, se a proposição é falsa.

ATENÇÃO: Antes de fazer a marcação, avalie cuidadosamente sua resposta.

LEMBRE-SE:

- A resposta correta vale 1 (um), isto é, você **ganha** 1 (um) ponto.
- A resposta errada vale -0,5 (menos meio ponto), isto é, você **não ganha** o ponto e ainda **tem descontada**, em outra questão que você acertou, essa fração do ponto.
- A ausência de marcação e a marcação dupla ou inadequada valem 0 (zero). Você **não ganha nem perde** nada.

2. Folha de Respostas

- A Folha de Respostas das Provas I e II e a Folha de Resposta da Redação são pré-identificadas. Confira os dados registrados nos cabeçalhos e assine-os com caneta esferográfica de **TINTA PRETA**, sem ultrapassar o espaço próprio.
- NÃO AMASSE, NÃO DOBRE, NÃO SUJE, NÃO RASURE ESSAS FOLHAS DE RESPOSTAS.
- Na Folha de Respostas destinada às Provas I e II, a marcação da resposta deve ser feita preenchendo-se o espaço correspondente com caneta esferográfica de **TINTA PRETA**. Não ultrapasse o espaço reservado para esse fim.

Exemplo de Marcação
na folha de Respostas

01	☐	F
02	☒	V
03	☒	V
04	☐	F
05	☒	V

- O tempo disponível para a realização das provas e o preenchimento das Folhas de Respostas é de 4 (quatro) horas e 30 (trinta) minutos.
-

ESTAS PROVAS DEVEM SER RESPONDIDAS PELOS CANDIDATOS AO SEGUINTE CURSO:

- SISTEMAS DE INFORMAÇÃO

PROVA I — ESTRUTURA DE DADOS

QUESTÕES de 01 a 35

INSTRUÇÃO:

Para cada questão, de **01** a **35**, marque na coluna correspondente da Folha de Respostas:

V, se a proposição é verdadeira;

F, se a proposição é falsa.

A resposta correta vale 1 (um ponto); a resposta errada vale -0,5 (*menos* meio ponto); a ausência de marcação e a marcação dupla ou inadequada valem 0 (zero).

Questão 01

Um algoritmo de complexidade **$O(1)$** (constante) é aquele em que não há crescimento do número de operações, pois ele independente do volume de dados de entrada (n), e, assim sendo, é o caso do acesso direto a um elemento de uma matriz, por exemplo.

Questão 02

Comparando-se 2 algoritmos, **A** e **B**, cujas complexidades são respectivamente **$O(n)$** e **$O(n \cdot \log(n))$** , pode-se afirmar que o algoritmo **A** é mais eficiente que o Algoritmo **B**.

Questão 03

Suponha um algoritmo **A** que possui 3 partes, **A_1** , **A_2** e **A_3** , com complexidade assintótica **$O(n)$** , **$O(n \cdot \log(n))$** e **$O(n^2)$** , respectivamente, pode-se afirmar que a complexidade assintótica de **A** é dada pela soma das complexidades assintóticas de **A_1** , **A_2** e **A_3** resultando em **$O(n^2 + n \cdot \log(n) + n)$** .

Questão 04

Em uma lista simplesmente encadeada, apenas a relação de sucessão é armazenada na estrutura de dados, através de um campo na estrutura do nó da lista que aponta para o próximo elemento e, com isso, os algoritmos de inserção e remoção, nesse tipo de lista, têm complexidade da ordem de **$O(n^2)$** .

QUESTÕES de 05 a 07

Para responder a essas questões, considere a seguinte definição de uma estrutura de dados em linguagem C.

```
typedef struct No
{
    float      dado;
    struct No* prox;
    struct No* ant;
} tNo;

typedef struct
{
    tNo* inicio;
    tNo* fim;
    int  numElems;
} tLista;
```

Questão 05

A função a seguir remove todos os nós da lista **tLista l**.

```
void limparLista(tLista *l) {
    tNo* aux = l->inicio;

    while(aux != NULL) {
        aux = aux->prox;
        free(l->inicio);
        l->inicio = aux;
    }
    l->inicio =
    l->ultimo = NULL;
}
```

Questão 06

A função indicada a seguir insere um novo nó no final da lista tLista l.

```
bool inserirElem(tLista *l, float k) {
    tNo* novoNo;

    novoNo = (tNo *)malloc(sizeof(tNo));

    if ( !novoNo )
        return false;

    novoNo->dado = k;
    novoNo->prox =
    novoNo->ant = NULL;

    if (l->inicio == NULL) {
        l->inicio =
        l->ultimo = novoNo;
        return true;
    }

    novoNo->ant      = l->ultimo;
    l->ultimo        = novoNo;
    return true;
}
```

Questão 07

A função indicada a seguir remove um nó da lista tLista l.

```
bool removerElem(tLista *l, float k) {
    tNo* noCor = l->inicio;

    while ( (noCor != NULL) && (noCor->dado != k) )
        noCor = noCor->prox;

    if (noCor == NULL)
        return false;

    if ( noCor->ant == NULL )
        l->inicio = noCor->prox;
    else
        noCor->ant->prox = noCor->prox;

    if ( noCor->prox == NULL )
        noCor->prox->ant = noCor->ant;
    else
        l->ultimo = noCor->ant;

    free(noCor);

    return true;
}
```

Questão 08

Uma **lista simplesmente encadeada circular** é caracterizada pelo último nó da lista apontar para o primeiro nó, e, portanto, é sempre necessário armazenar na estrutura da lista, além do ponteiro para seu primeiro nó, um ponteiro para o seu último nó, uma vez que, caso contrário, não será possível identificar o último elemento, pois seu campo próximo não apontará para NULL, e sim para o primeiro elemento da lista.

Questão 09

A política de acesso **LIFO** é aplicada em uma estrutura de dados toda a vez que se quer garantir que os dados inseridos na estrutura sejam retirados na ordem inversa, sendo esse o caso da estrutura de dados do tipo **Pilha**.

Questão 10

A implementação da estrutura de dados **Pilha** pode ser feita a partir de um vetor, e, com isso, as operações de **Empilhar** e **Desempilhar** terão, ambas, complexidade da ordem de **O(1)**.

Questão 11

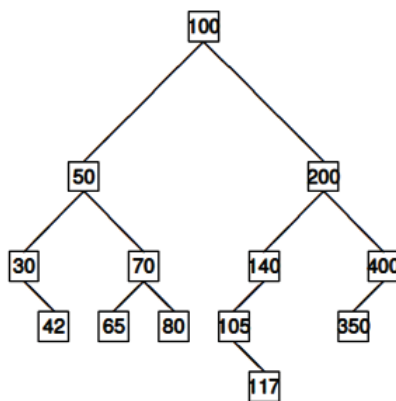
A estrutura de dados do tipo **Fila** implementa uma política de acesso **FIFO**, o que implica que as operações de **Enfileirar** e **Desenfileirar** elementos devem ser feitas sempre em extremidades opostas da estrutura, ou seja, **Enfileirar** no final e **Desenfileirar** do início.

Questão 12

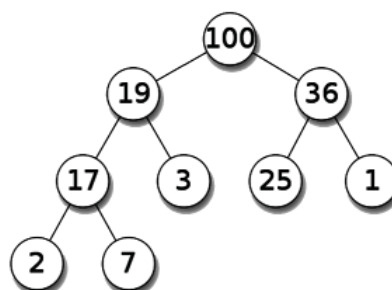
Um algoritmo de complexidade **O(n²)** (quadrático) é factível, mas tende a se tornar muito ruim quando a quantidade de dados é suficientemente grande, sendo o caso de algoritmos que têm dois laços encadeados como o processamento de itens em uma matriz bidimensional.

QUESTÕES de 13 a 18

Para responder a essas questões, considere as árvores binárias:



(a)



(b)

Questão 13

A árvore da figura (a) tem **altura** igual a 4.

Questão 14

A árvore da figura (a) é **estritamente binária**.

Questão 15

Aplicando-se o **percurso em ordem simétrica (in-order)** na **árvore binária** da figura (a), tem-se a seguinte sequência de visitação dos seus nós: 30, 42, 50, 65, 70, 80, 100, 105, 117, 140, 200, 350, 400.

Questão 16

A árvore da figura (b) é uma **Max-Heap**.

Questão 17

A complexidade do algoritmo de **inserção** de um novo nó em uma **Heap** é da ordem de **O(n)**, no pior caso.

Questão 18

A complexidade do algoritmo de **remoção** de um nó em um **Heap** é da ordem de **O(n.log(n))**, no pior caso.

Questão 19

O **algoritmo de ordenação por inserção** é indicado para conjuntos de valores parcialmente ordenados, ou seja, onde apenas uma parte dos elementos se encontra ordenados.

Questão 20

O **algoritmo de ordenação Quicksort** tem sempre complexidade **$O(n \cdot \log(n))$** para qualquer configuração de entrada.

Questão 21

Alguns algoritmos de ordenação têm como **melhor caso** a situação do conjunto de entrada já se encontrar ordenado, não sendo esse o caso do **algoritmo de ordenação por Seleção**, que tem complexidade **$O(n^2)$** .

Questão 22

Alguns algoritmos de ordenação têm como **pior caso** a situação em que o conjunto de entrada está ordenado na ordem inversa à da ordenação pretendida, ou seja, em ordem crescente, quando a ordenação pretendida é decrescente, por exemplo, sendo o que ocorre no **algoritmo Heapsort**, que, nesse caso, possui complexidade **$O(n^2)$** .

Questão 23

O processo de **pivoteamento** no algoritmo **Quicksort** consiste em escolher um elemento do conjunto a ser ordenado, o **pivô**, que irá dividir o conjunto em duas partes: maiores e menores que o **pivô** respectivamente, e, para garantir o melhor caso do algoritmo, é necessário que as duas partes tenham o mesmo número de elementos, o que acontece quando o **pivô** corresponde ao valor médio do conjunto de entrada.

Questão 24

Algoritmos de **ordenação externa baseados em intercalação** têm como parâmetros o número de blocos de registros que a memória interna comporta e o número máximo de arquivos sequenciais (**fitas**) que se pode ter para leitura e escrita dos blocos ordenados.

Questão 25

Em um **algoritmo de ordenação externa por intercalação com f-caminhos**, são necessários **$2f$** arquivos sequenciais (**fitas**) para uma ordenação eficiente.

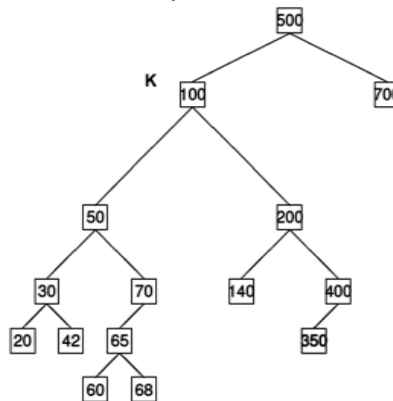
Questão 26

Em um **algoritmo de ordenação externa por intercalação com f-caminhos**, é possível reduzir o número de arquivos sequenciais (**fitas**) de **$2f$** para **$f+1$** , ao custo de uma passada adicional de redistribuição dos blocos da fita $f+1$ para cada uma das f fitas, a cada intercalação.

RASCUNHO

QUESTÕES 27 e 28

Considere a figura para responder a essas questões



Questão 27

Aplicando-se o algoritmo de inserção de nós de uma **árvore binária de busca** a partir do conjunto de chaves $C = \{ 500, 700, 100, 50, 30, 20, 70, 65, 400, 200, 60, 68, 42, 140, 350 \}$, produz-se como resultado a árvore da figura.

Questão 28

A remoção da chave **100**, localizada no nó **K** da árvore da figura, implica a busca nas duas subárvores de **K**, por uma folha que contenha uma chave que possa substituir a chave do nó **K**, mantendo a condição da árvore binária de busca, e, portanto, esse processo de busca deve verificar todos os nós folhas de suas duas subárvores, procurando pelo nó **M**, que deve conter a maior ou menor chave, sendo feita, então, a troca entre as chaves do nó folha **M** e do nó **K**, e, em seguida, o nó folha, agora contendo o valor 100, é removida da árvore.

Questão 29

Uma **árvore binária balanceada** é dita **AVL**, se, para qualquer um de seus nós, as alturas de suas duas subárvores, esquerda e direita, diferem, em módulo, de até uma unidade.

Questão 30

Toda **árvore balanceada** tem altura da ordem de $O(\log(n))$.

Questão 31

Em uma **árvore AVL**, após a inserção de um novo nó, cada um de seus nós é verificado quanto à condição de balanceamento, e, caso a condição falhe, uma operação de **rotação simples à direita** ou **rotação simples à esquerda** é aplicada, corrigindo o desbalanceamento de todos os nós desbalanceados.

Questão 32

Os métodos de **busca baseados em transformações de chaves (hashing)** possuem complexidade média da ordem de $O(1)$.

Questão 33

Nos métodos de **busca baseados em transformações de chaves**, a **função de hashing** ideal é aquela que é simples de ser computada e, para cada chave de entrada, qualquer uma das saídas possíveis é igualmente provável de ocorrer, de modo que, com isso espera-se minimizar as colisões de chaves.

Questão 34

Uma **árvore B de ordem k** é uma árvore onde todos os nós (ou páginas) contêm mais de uma chave: no máximo, $2k$ chaves e, no mínimo, k chaves, e, todas as páginas folhas estão no mesmo nível da árvore.

Questão 35

A operação de **inserção** de uma nova chave em uma **árvore B de ordem k** pode envolver a “explosão” (*split*) de um nó, caso o seu limite de chaves seja ultrapassado (*overflow*), e, no pior caso, em que os *splits* acontecem em sequência, a complexidade da inserção na árvore B é da ordem de $O(n^2)$.

PROVA II — CIRCUITOS DIGITAIS E ARQUITETURA DE COMPUTADORES

QUESTÕES de 36 a 70

INSTRUÇÃO:

Para cada questão, de **36** a **70**, marque na coluna correspondente da Folha de Respostas:

V, se a proposição é verdadeira;

F, se a proposição é falsa.

A resposta correta vale 1 (um ponto); a resposta errada vale -0,5 (*menos* meio ponto); a ausência de marcação e a marcação dupla ou inadequada valem 0 (zero).

Questão 36

A expressão $(011011)_2 + (001101)_2$ gera como resultado, com 6 bits, o valor $(40)_{10}$.

Questão 37

A expressão $(1AF)_{16} - (2B)_{16} - (120)_{16}$ gera como resultado o valor $(1000000)_2$ com 7 bits de representação.

Questão 38

O número binário de 10 bits $(0110100000)_2$ convertido para a base decimal é igual a $(380)_{10}$.

Questão 39

O número decimal -72 representado em complemento de 2 (C2), com 8 bits, é $(01001000)_2$.

Questão 40

O número decimal -128, representado com 9 bits, nas formas sinal-magnitude e complemento de 2 (C2) é igual a $(110000000)_2$ em ambas representações.

Questão 41

No padrão IEEE 754 de representação de números binários em ponto flutuante com 32 bits, o expoente é considerado polarizado quando for representado com 8 bits, sendo o bit mais significativo (MSB) sempre igual a zero.

Questão 42

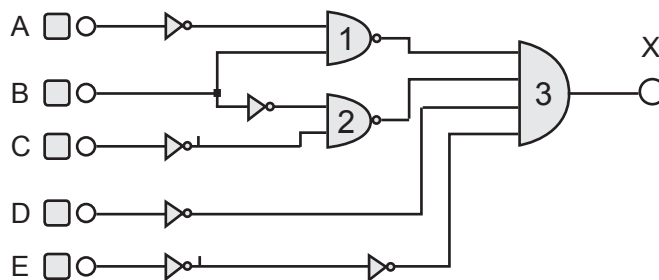
O padrão IEEE 754 de representação de números binários em ponto flutuante com 64 bits, (precisão dupla) utiliza um bit para o sinal, 11 bits para o expoente e 52 bits para a mantissa.

Questão 43

O número binário 1001011001 corresponde ao número 1101110101 em código Gray.

RASCUNHO

QUESTÕES 44 e 45



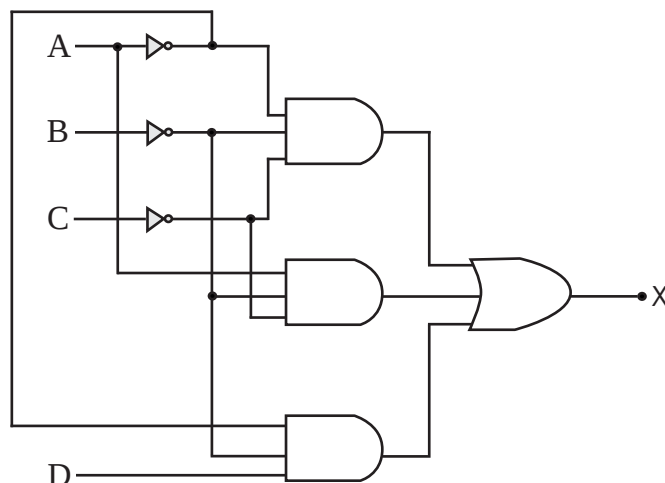
Questão 44

Considerando o circuito formado por duas portas NAND (1 e 2) e uma porta AND (3), com os respectivos valores das entradas $A=0$, $B=1$, $C=1$, $D=0$ e $E=1$, o valor produzido na saída X é igual a 1.

Questão 45

A supressão de todas as portas NOT do circuito, mantendo-se as portas NAND (1 e 2) inalteradas, gera como entradas para a porta AND (3), os bits 0101, de cima (saída da porta NAND 1) para baixo, se são aplicados os valores de entrada: $A=0$, $B=1$, $C=1$, $D=0$ e $E=1$.

Questão 46



A expressão algébrica não simplificada, na forma de soma de produtos, para o circuito apresentado, é $(!A.!B.!C) + (A.!B.!C) + (!A.!B.D)$, sendo o ponto de exclamação ! a operação NOT.

Questão 47

Considerando a expressão algébrica $S = (A.!B.!C) + (A.!B.C) + (A.B.C)$, na qual o ponto de exclamação é a operação NOT, e a aplicação das regras distributiva (fatoração) e de absorção da Álgebra Booleana, é correto afirmar que uma possível simplificação para essa expressão é $S = A.(!B+C)$.

RASCUNHO

Questão 48

A aplicação dos teoremas e das regras da Álgebra Booleana, incluindo os Teoremas de DeMorgan, à expressão não simplificada $Z = !((!A.C) + (B + !D))$, gera como uma possível saída a expressão simplificada $Z = (A.!C) + (!B.D)$.

QUESTÕES 49 e 50

A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

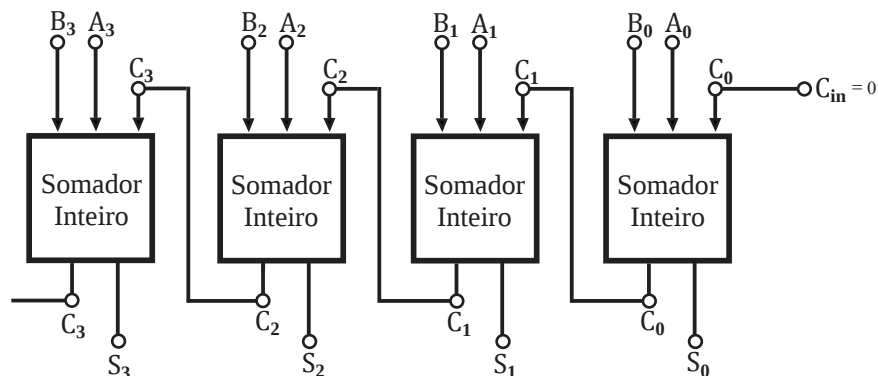
Questão 49

A expressão de produto de somas para a tabela-verdade apresentada, considerando as entradas A, B e C e a saída S, é composta pelos maxterms M_0 , M_2 , M_5 e M_6 .

Questão 50

Um mapa de Karnaugh-Veitch, para as saídas em nível ALTO ($S = 1$) da tabela-verdade apresentada, gera a expressão algébrica $S = (A.B) + (B.C) + (A.C)$.

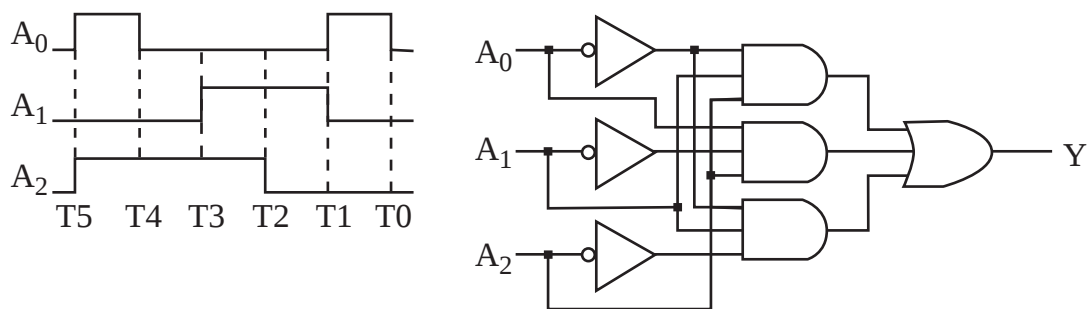
Questão 51



Na operação do somador completo, considerando como valores de entrada $A_3A_2A_1A_0$ e $B_3B_2B_1B_0$, respectivamente, 1101 e 0111 e $carry\ in\ (C_{in}) = 0$, as saídas $S_3S_2S_1S_0$ são, respectivamente, 0100, sendo o $carry\ out\ C_3$ igual a 1.

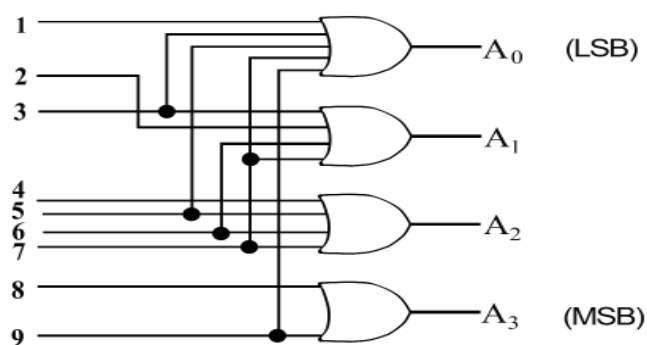
RASCUNHO

Questão 52



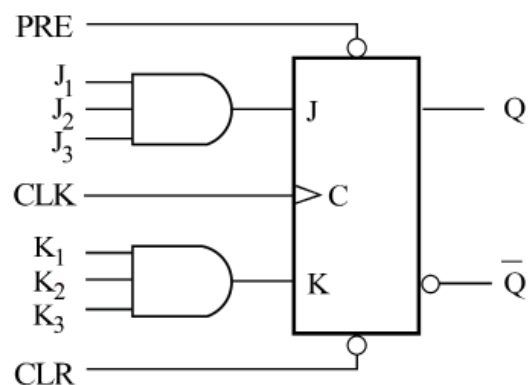
Considerando as formas de onda submetidas às entradas do circuito, no instante de observação T_2 é gerada a saída 0.

Questão 53



No codificador de decimal para BCD mostrado na figura, quando a entrada 7 é ligada (recebendo bit 1) e as demais entradas são desligadas (recebendo bit 0), o valor gerado nas saídas $A_0A_1A_2A_3$ é 0110, respectivamente.

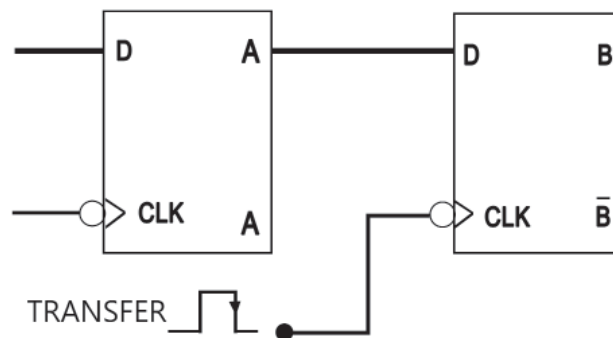
Questão 54



Um flip-flop do tipo J-K gera saídas válidas (Q e $\bar{Q}$) para cada uma das 4 combinações possíveis dos valores de J e K : 00, 01, 10 e 11.

RASCUNHO

Questão 55



A figura ilustra a utilização de dois flip-flops D, que empregam transferência síncrona e, nessa configuração, o valor armazenado no flip-flop da esquerda é transferido para o flip-flop da direita (ligação $A \rightarrow D$), na borda de descida do pulso TRANSFER.

Questão 56

O conceito de programa armazenado e o gargalo de von Neumann representam, respectivamente, uma contribuição e uma limitação (ou ponto fraco) do modelo de arquitetura conhecido como “máquina de von Neumann”.

QUESTÕES 57 e 58

Registradores			
R0	R1	R2	R3
0	110	1110	120

Memória	
Endereço (Posição)	Conteúdo
100	0
110	200
120	100
...	
200	1200
...	
1100	100
1110	300
...	
1200	200

Questão 57

Considerando uma arquitetura que utilize o modo de endereçamento direto por registrador e o formato de instrução <OPCODE OP1,OP2,OP3>, sendo OPCODE o código (mnemônico) da instrução, OP1 o operando-destino e OP2 e OP3 os operandos-fonte, então a instrução <ADD R0,R1,R2> atribui ao registrador R0 o valor 500.

RASCUNHO

Questão 58

Considerando uma arquitetura que utilize o modo de endereçamento indireto por registrador e o formato de instrução <OPCODE OP1,OP2>, sendo OPCODE o código (mnemônico) da instrução, OP1 o operando-destino e o OP2 o operando-fonte, então a instrução <MOV R0,R3> atribui ao registrador R0 o valor 100.

Questão 59

As arquiteturas RISC (*reduced instruction-set computers*) caracterizam-se por empregarem conjuntos de instruções menores (com menos instruções) e mais otimizados quando comparadas às arquiteturas CISC (*complex instruction-set computers*).

QUESTÕES 60 e 61

Número da instrução	Código da instrução (OPCODE)	Operando 1	Operando 2	Operando 3	Descrição
I1	LW	\$t0	0(\$t0)		// \$t0 = memória[0 + \$t0]
I2	LW	\$t1	20(\$t1)		// \$t1 = memória[20 + \$t1]
I3	ADD	\$s3	\$t0	\$t1	// \$s3 = \$t0 + \$t1
I4	MOV	\$t0	\$s3		// \$t0 = \$s3
I5	SUB	\$s3	\$s4	\$t1	// \$s3 = \$s4 - \$t1
I6	ADDI	\$t1	\$s3	200	// \$t1 = \$s3 + 200

LW = load word, ADD = soma, SUB = subtração, ADDI = add immediate, MOV = move (cópia entre registradores). \$t0, \$t1, \$s3 e \$s4: registradores.

Questão 60

Considerando o trecho de código, pode-se afirmar que as instruções I1, I2 e I4, por não usarem o campo Operando 3, possuem um tamanho de instrução menor do que as demais instruções.

Questão 61

O trecho de código apresentado possui uma dependência verdadeira (RAW – *read after write*) entre as instruções I2 e I3, no acesso ao registrador \$t1.

Questão 62

Arquiteturas superescalares implementam técnicas avançadas de *pipelining* para serem capazes de iniciar múltiplas instruções em um mesmo ciclo de processador.

Questão 63

O mecanismo de interrupções implementado pelo hardware pode ser desabilitado sem prejuízo de operações básicas do hardware, tais como acesso a discos (entrada e saída) e troca de contextos entre processos em execução.

RASCUNHO

Questão 64

A principal diferença entre processadores com um único núcleo (*monoprocessadores*) e processadores com múltiplos núcleos (*multicore*) está na capacidade de execução de múltiplas tarefas simultaneamente (*multitarefa*), presente somente no segundo tipo (*multicore*).

Questão 65

Uma das principais características das unidades de processamento gráfico (GPUs) é a quantidade significativamente maior de elementos de processamento (ou processadores), quando comparadas ao processadores com múltiplos núcleos (*multicore*).

Questão 66

As unidades de processamento gráfico (GPU) são capazes de executar uma determinada aplicação sem a necessidade de executar instruções no processador (CPU).

Questão 67

No contexto dos processadores, o conceito de vazão (ou *throughput*) corresponde à quantidade de instruções que um processador consegue executar dentro de um determinado intervalo de tempo (um segundo, por exemplo).

Questão 68

A utilização é uma métrica usada para medir a fração de tempo em que um processador permanece ocupado atendendo às requisições feitas pelos programas ou usuários.

Questão 69

Se um computador A executa um programa em 10 segundos e um computador B executa esse mesmo programa em 15 segundos, é correto afirmar que o computador A é duas vezes mais rápido que o computador B.

Questão 70

$$\text{TempoCPU} = \frac{\text{Quantidade de instruções do programa} \times \text{CPI}}{\text{Frequência do processador (clock rate)}}$$

A equação clássica para o cálculo do desempenho do processador (TempoCPU) envolve os três parâmetros que afetam diretamente este desempenho, dos quais o CPI (*clock cycles per instruction*) representa a quantidade média de ciclos de processador usada para a execução de cada instrução.

RASCUNHO

PROVA DE REDAÇÃO

INSTRUÇÕES:

- Escreva sua Redação com caneta de tinta AZUL ou PRETA, de forma clara e legível.
- Caso utilize letra de imprensa, destaque as iniciais maiúsculas.
- O rascunho deve ser feito no local apropriado do Caderno de Questões.
- Na Folha de Resposta, utilize apenas o espaço a ela destinado.
- Será atribuída a pontuação ZERO à Redação que
 - se afastar do tema proposto;
 - for apresentada em forma de verso;
 - for assinada fora do local apropriado;
 - apresentar qualquer sinal que, de alguma forma, possibilite a identificação do candidato;
 - for escrita a lápis, em parte ou na sua totalidade;
 - apresentar texto incompreensível ou letra ilegível.

Os textos a seguir devem servir como ponto de partida para a sua Redação.

I.

[...] Com algum exagero, quase se pode afirmar que *Raízes do Brasil* não está completando oitenta anos: o livro que gerações de leitores conheceram é, na verdade, de 1948.

Antes de falar no sentido dessa mudança, é preciso delinear, de forma breve, que livro afinal é este. Ensaio enxuto, com menos de 200 páginas, *Raízes do Brasil* compõe um concentrado painel interpretativo da história do Brasil, identificando certos traços fortes da formação nacional. Nos quatro primeiros capítulos, o colonizador português faz um herói ambíguo. Para Sérgio Buarque, os portugueses eram os “portadores naturais” de uma “missão histórica”: a “conquista do trópico para a civilização”. Adaptáveis às condições hostis da natureza e desprovidos de orgulho racial, eles cultivavam um espírito relaxado e aventureiro, que, com a exploração da mão de obra escrava, se provaria eficiente na América. O personalismo ibérico, de outro lado, encontrou terreno próprio na grande propriedade rural, onde a voz do proprietário e patriarca era lei. Desse caldo de cultura aquecido ao sol do Novo Mundo, emerge o tipo social que, com certa ironia, Sérgio Buarque qualifica de “contribuição brasileira para a civilização”: o homem cordial.

TEIXEIRA, J. Clássicos em mutação. **Veja**, ed. 2491, ano 49, n. 33, São Paulo: Abril, p. 84, 17 ago. 2016.

II.

Um fascinante mal-entendido tem assombrado a história cultural brasileira nas últimas oito décadas. Em 1936, ao publicar seu livro de estreia, Sérgio Buarque de Holanda teria identificado o perfil da identidade nacional: a cordialidade. No entanto, para o leitor da obra, essa associação desinibida surpreende. No fundo, *Raízes do Brasil* é um ensaio-manifesto contra a ideia de cordialidade. Sérgio Buarque desenvolveu o conceito para dar conta da formação social brasileira nos séculos nos quais o mundo agrário era dominante. Ao mesmo tempo, ele apostou suas fichas no universo urbano e industrializado, que, em tese, deveria varrer o homem cordial do mapa. No passado agrário, a família patriarcal ditava o tom das relações, forjando uma sociabilidade sujeita aos privilégios deste ou daquele grupo, em lugar de investir num projeto coletivo, corporificado na metáfora do espaço público. [...]

Em *Raízes do Brasil*, a cordialidade não é um traço exclusivamente nacional. Por isso, na imaginação crítica de Sérgio Buarque, a abolição e a urbanização condenariam o homem cordial ao museu da história do Brasil – ruína do passado agrário, a ser devidamente superada pela modernização. Esse é o sentido forte de sua resposta a Cassiano Ricardo: “O homem cordial se acha fadado a desaparecer, onde ainda não desapareceu de todo. E, às vezes, receio sinceramente que já tenha gasto muita cera com esse pobre defunto”. Palavras duras, escritas em 1948, e que esclarecem o tropeço dos que veem no conceito mais uma das perversas maquinações da elite econômica para inventar uma “identidade nacional”, a fim de ocultar desigualdade e injustiças.

TEIXEIRA, J. Clássicos em mutação. **Veja**, ed. 2491, ano 49, n. 33, São Paulo: Abril, p. 86-87, 17 ago. 2016.

III.

A forma como a atual cena política brasileira se apresenta, em meio à propagação de discursos reacionários, parece colocar uma rasura nas ideias da gentileza e respeito às diferenças com as quais o brasileiro costuma ver o próprio país. Uma rasura que remete à ideia do homem cordial, forjada no livro *Raízes do Brasil* (1936), onde o historiador Sérgio Buarque de Holanda (1902-1982) debruça-se sobre as origens da cordialidade nacional.

Teresa Santana, historiadora que assinou o artigo *O nosso fundamentalismo* (2013), confeccionado nas barbas das manifestações de junho de 2013, as maiores desde a redemocratização nacional, fala em “momento apropriado para repensar o caráter do brasileiro”. “Afirmar que somos naturalmente tolerantes é desconhecer o machismo, a homofobia e o racismo que vigoram nos trens, ônibus e vagões lotados. No fundo, se não repensarmos nosso caráter, estaremos condenados a ser uma sociedade autista”.

REZENDE, E. O homem cordial. **Muito**, #417, Salvador, p. 15, 3 jul. 2016. Revista do Grupo A Tarde.

PROPOSTA

Com base nas ideias dos fragmentos em destaque e também nas suas próprias vivências, escreva **um texto argumentativo** em que você discuta criticamente o pensamento da historiadora Teresa Santana: “**Afirmar que somos tolerantes é desconhecer o machismo, a homofobia e o racismo. Se não repensarmos nosso caráter, seremos uma sociedade autista.**”

RASCUNHO

RASCUNHO



UNIVERSIDADE FEDERAL DA BAHIA
PROGRAD/COORDENAÇÃO DE SELEÇÃO E ORIENTAÇÃO
Rua Dr. Augusto Viana, 33 – Canela
Cep. 40110-060 – Salvador/BA
Telefax (71) 3283-7820 – E-mail: ssoa@ufba.br
Site: www.vagasresiduais.ufba.br