

Análise de Algoritmo

Exercício

Com base nos conceitos de teoria da complexidade, avalie as afirmações a seguir.

- I. A equação de recorrência que define a complexidade da função F é a mesma do algoritmo clássico de ordenação mergesort.
- II. O número de chamadas recursivas da função F é $\Theta(\log n)$.
- III. O número de vezes que a função G da linha 4 é chamada é $O(n \log n)$.

É correto o que se afirma em

- (A) I, apenas.
- (B) II, apenas.
- (C) I e III, apenas.
- (D) II e III, apenas.
- (E) I, II e III.

Com base nos conceitos de teoria da complexidade, avalie as afirmações a seguir.

- I. A equação de recorrência que define a complexidade da função F é a mesma do algoritmo clássico de ordenação mergesort.
- II. O número de chamadas recursivas da função F é $\Theta(\log n)$.
- III. O número de vezes que a função G da linha 4 é chamada é $O(n \log n)$.

```
1 void F(int n) {
2     if(n > 0) {
3         for(int i = 0; i < n; i++) {
4             G(i);
5         }
6         F(n/2);
7     }
8 }
```

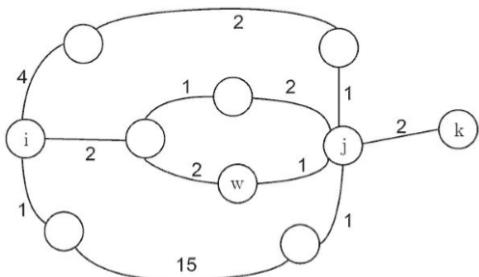
Letra D

Exercício

A figura a seguir exibe um grafo que representa um mapa rodoviário, no qual os vértices representam cidades e as arestas representam vias. Os pesos indicam o tempo atual de deslocamento entre duas cidades.

Considerando que os tempos de ida e volta são iguais para qualquer via, avalie as afirmações a seguir acerca desse grafo.

- I. Dado o vértice de origem i, o algoritmo de Dijkstra encontra o menor tempo de deslocamento entre a cidade i e todas as demais cidades do grafo.
- II. Uma árvore geradora de custo mínimo gerada pelo algoritmo de Kruskal contém um caminho de custo mínimo cuja origem é i e cujo destino é k.
- III. Se um caminho de custo mínimo entre os vértices i e k contém o vértice w, então o subcaminho de origem w e destino k deve também ser mínimo



É correto o que se afirma em

- (A) I, apenas.
- (B) II, apenas.
- (C) I e III, apenas.
- (D) II e III, apenas.
- (E) I, II e III.

Letra E

Exercício

Um país utiliza moedas de 1, 5, 10, 25 e 50 centavos. Um programador desenvolveu o método a seguir, que implementa a estratégia gulosa para o problema do troco mínimo. Esse método recebe como parâmetro um valor inteiro, em centavos, e retorna um array no qual cada posição indica a quantidade de moedas de cada valor.

```
public static int[] troco(int valor){  
    int[] moedas = new int[5];  
  
    moedas[4] = valor / 50;  
    valor = valor % 50;  
    moedas[3] = valor / 25;  
    valor = valor % 25;  
    moedas[2] = valor / 10;  
    valor = valor % 10;  
    moedas[1] = valor / 5;  
    valor = valor % 5;  
    moedas[0] = valor;  
    return moedas;  
}
```

Considerando o método apresentado, avalie as asserções a seguir e a relação proposta entre elas.

I. O método guloso encontra o menor número de moedas para o valor de entrada, considerando as moedas do país.

PORQUE

II. Métodos gulosos sempre encontram a solução ótima global.

A respeito dessas asserções, assinale a opção correta.

- (A) As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
- (B) As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.
- (C) A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- (D) A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- (E) As asserções I e II são proposições falsas.

Letra C.

Exercício

A sequência de Fibonacci é uma sequência de números inteiros que começa em 1, a que se segue 1, e na qual cada elemento subsequente é a soma dos dois elementos anteriores. A função fib a seguir calcula o n-ésimo elemento da sequência de Fibonacci:

```
unsigned int fib (unsigned int n)  
{  
    if (n < 2)  
        return 1;  
    return fib(n - 2) + fib(n - 1);  
}
```

Considerando a implementação acima, avalie as afirmações a seguir.

- I. A complexidade de tempo da função fib é exponencial no valor de n.
- II. A complexidade de espaço da função fib é exponencial no valor de n.
- III. É possível implementar uma versão iterativa da função fib com complexidade de tempo linear no valor de n e complexidade de espaço constante.

É correto o que se afirma em

- (A) I, apenas.
- (B) II, apenas.
- (C) I e III, apenas.
- (D) II e III, apenas.
- (E) I, II e III.

Letra C

Banco de dados

Questão 23 - ENADE 2008

Considere o esquema de banco de dados relacional apresentado a seguir, formado por 4 relações, que representa o conjunto de estudantes de uma universidade que podem, ou não, morar em repúblicas (moradias compartilhadas por estudantes). A relação Estudante foi modelada como um subconjunto da relação Pessoa. Considere que os atributos grifados correspondam à chave primária da respectiva relação e os atributos que são seguidos da palavra **referencia** sejam chaves estrangeiras.

```
Pessoa (IdPessoa:integer, Nome:varchar(40), Endereco:varchar(40))
FonePessoa (IdPessoa:integer referencia Pessoa, DDD:varchar(3), Prefixo:char(4), Nro:char(4))
Republica (IdRep:integer, Nome:varchar(30), Endereco:varchar(40))
Estudante (RA:integer, Email:varchar(30), IdPessoa:integer referencia Pessoa, IdRep:integer referencia Republica)
```

Suponha que existam as seguintes tuplas no banco de dados:

```
Pessoa(1, 'José Silva', 'Rua 1, 20');
República(20, 'Várzea', 'Rua Chaves, 2001')
```

Qual opção apresenta apenas tuplas válidas para esse esquema de banco de dados relacional?

- A Estudante(10, 'jsilva@ig.com.br', null, 20); FonePessoa(10, '019', '3761', '1370')
- B Estudante(10, 'jsilva@ig.com.br', 1, null); FonePessoa(10, '019', '3761', '1370')
- C Estudante(10, 'jsilva@ig.com.br', 1, 20); FonePessoa(1, null, '3761', '1370')
- D Estudante(10, 'jsilva@ig.com.br', 1, 50); FonePessoa(1, '019', '3761', '1370')
- E Estudante(10, 'jsilva@ig.com.br', 1, null); FonePessoa(1, '019', '3761', '1370')

Questão 46 - ENADE 2011

Em um modelo de dados que descreve a publicação acadêmica de pesquisadores de diferentes instituições em eventos acadêmicos, considere as tabelas abaixo.

DEPARTAMENTO (#CodDepartamento, NomeDepartamento)

EMPREGADO (#CodEmpregado, NomeEmpregado, CodDepartamento, Salario)

Na linguagem SQL, o comando mais simples para recuperar os códigos dos departamentos cuja média salarial seja maior que 2000 é

A SELECT CodDepartamento
FROM EMPREGADO
GROUP BY CodDepartamento
HAVING AVG (Salario) > 2000

B SELECT CodDepartamento
FROM EMPREGADO
WHERE AVG (Salario) > 2000
GROUP BY CodDepartamento

C SELECT CodDepartamento
FROM EMPREGADO
WHERE AVG (Salario) > 2000

D SELECT CodDepartamento, AVG (Salario) > 2000
FROM EMPREGADO
GROUP BY CodDepartamento

E SELECT CodDepartamento
FROM EMPREGADO
GROUP BY CodDepartamento
ORDER BY AVG (Salario) > 2000

Questão 73 - ENADE 2005

Considere o seguinte script SQL de criação de um banco de dados.

```
CREATE TABLE PEÇAS (CODIGO NUMERIC(5) NOT NULL,
DESCRICAOP VARCHAR(20) NOT NULL,
ESTOQUE NUMERIC(5) NOT NULL,
PRIMARY KEY(CODIGO));

CREATE TABLE FORNECEDORES
(COD_FORN NUMERIC(3) NOT NULL,
NOME VARCHAR(30) NOT NULL,
PRIMARY KEY(COD_FORN));

CREATE TABLE FORNECIMENTOS
(COD_PECA NUMERIC(5) NOT NULL,
COD_FORN NUMERIC(3) NOT NULL,
QUANTIDADE NUMERIC(4) NOT NULL,
PRIMARY KEY(COD_PECA, COD_FORN),
FOREIGN KEY (COD_PECA) REFERENCES PEÇAS,
FOREIGN KEY (COD_FORN) REFERENCES FORNECEDORES);
```

A partir desse script, assinale a opção que apresenta comando SQL que permite obter uma lista que contenha o nome de cada fornecedor que tenha fornecido alguma peça, o código da peça fornecida, a descrição dessa peça e a quantidade fornecida da referida peça.

- A SELECT * FROM PEÇAS, FORNECEDORES, FORNECIMENTOS;
- B SELECT * FROM PEÇAS, FORNECEDORES, FORNECIMENTOS WHERE PEÇAS.CODIGO = FORNECIMENTOS.COD_PECA AND FORNECEDORES.COD_FORN = FORNECIMENTOS.COD_FORN;
- C SELECT NOME, CODIGO, DESCRICAOP, QUANTIDADE FROM PEÇAS, FORNECEDORES, FORNECIMENTOS;
- D SELECT NOME, CODIGO, DESCRICAOP, QUANTIDADE FROM PEÇAS, FORNECEDORES, FORNECIMENTOS WHERE PEÇAS.CODIGO = FORNECIMENTOS.COD_PECA AND FORNECEDORES.COD_FORN = FORNECIMENTOS.COD_FORN;
- E SELECT DISTINCT NOME, CODIGO, DESCRICAOP, QUANTIDADE FROM PEÇAS, FORNECEDORES, FORNECIMENTOS WHERE CODIGO = COD_PECA;

Compiladores

Em um compilador, um analisador sintático descendente preditivo pode ser implementado com o auxílio de uma tabela construída a partir de uma gramática livre de contexto. Essa tabela, chamada tabela LL(k), indica a regra de produção a ser aplicada olhando-se o k -ésimo próximo símbolo lido, chamado *lookahead*(k). Por motivo de eficiência, normalmente busca-se utilizar $k=1$. Considere a gramática livre de contexto $G = (X, Y, Z, a, b, c, d, e, P, X)$, em que P é composto pelas seguintes regras de produção:

$$\begin{aligned} X &\rightarrow aZbXY \mid c \\ Y &\rightarrow dX \mid \epsilon \\ Z &\rightarrow e \end{aligned}$$

Considere, ainda, a seguinte tabela LL(1), construída a partir da gramática G , sendo $\$$ o símbolo que representa o fim da cadeia. Essa tabela possui duas produções distintas na célula (Y, d) , gerando, no analisador sintático, uma dúvida na escolha da regra de produção aplicada em determinados momentos da análise.

	a	b	c	d	e	\$
X	$X \rightarrow aZbXY$		$X \rightarrow c$			
Y				$Y \rightarrow dX$ $Y \rightarrow \epsilon$		$Y \rightarrow \epsilon$
Z						$Z \rightarrow e$

Considerando que o processo da construção dessa tabela LL(1), a partir da gramática G , foi seguido corretamente, a existência de duas regras de produção distintas na célula (Y, d) , neste caso específico, resulta

- (A) da ausência do símbolo de fim de cadeia ($\$$) nas regras de produção.
- (B) de um não-determinismo causado por uma ambiguidade na gramática.
- (C) do uso incorreto do símbolo de cadeia vazia (ϵ) nas regras de produção.
- (D) da presença de duas regras de produção com um único terminal no corpo.
- (E) da presença de duas regras de produção com o mesmo não terminal na cabeça.

Letra B

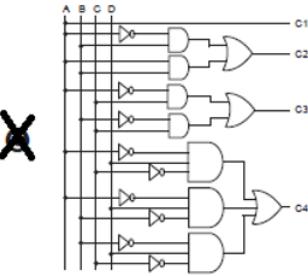
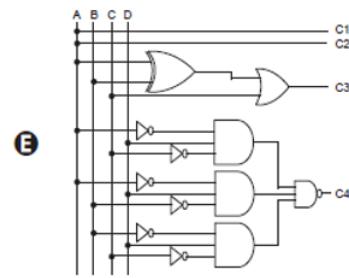
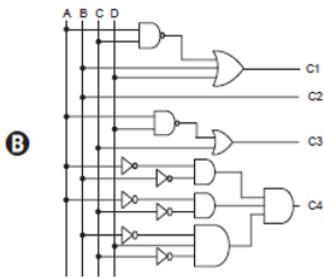
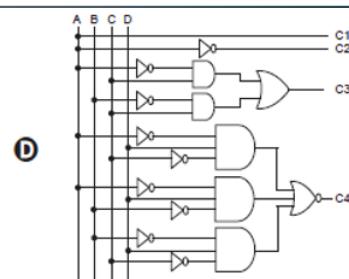
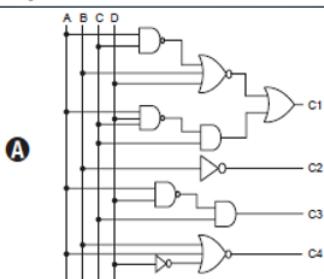
Eletrônica Digital

Exercício

Considere que o gestor do restaurante de uma universidade pretenda implantar um sistema de catracas eletrônicas utilizando circuitos combinatórios para o acesso dos clientes à sala de alimentação. O funcionamento básico desse sistema é descrito a seguir.

- Deve haver quatro filas de espera: A, B, C e D;
- deve haver sensores que indiquem a presença de clientes nas filas, sendo atribuído o valor 1 quando houver cliente na fila, e 0 quando a fila estiver vazia;
- quando determinada fila estiver vazia, sua catraca deverá ficar travada, e as filas devem estar na ordem de prioridade a seguir: A > B > C > D;
- no máximo, duas catracas poderão estar abertas em um mesmo instante, respeitando-se a ordem de prioridade;
- as catracas das filas A, B, C e D devem ser representadas pelas variáveis C1, C2, C3 e C4, respectivamente;
- o valor de saída 1 deverá indicar que a catraca está aberta, e o valor 0, que a catraca está fechada.

Nesse contexto, qual circuito combinatório deverá resolver o problema de acesso dos clientes à sala de alimentação do restaurante da universidade?

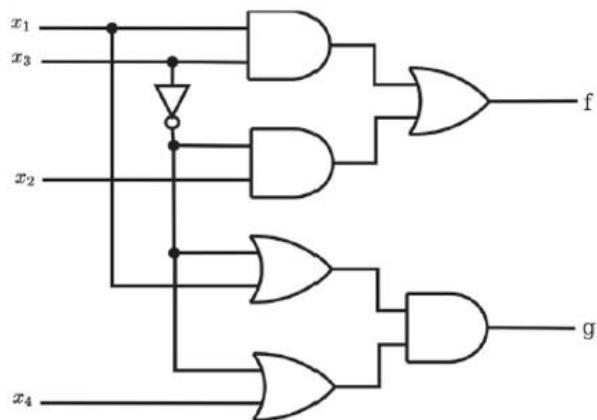


Exercício

A tabela a seguir apresenta a relação de mintermos e maxtermos para três variáveis.

Linha	x_1	x_2	x_3	Mintermo	Maxtermo
0	0	0	0	$m_0 = \bar{x}_1 \bar{x}_2 \bar{x}_3$	$M_0 = x_1 + x_2 + x_3$
1	0	0	1	$m_1 = \bar{x}_1 \bar{x}_2 x_3$	$M_1 = x_1 + x_2 + \bar{x}_3$
2	0	1	0	$m_2 = \bar{x}_1 x_2 \bar{x}_3$	$M_2 = x_1 + \bar{x}_2 + x_3$
3	0	1	1	$m_3 = \bar{x}_1 x_2 x_3$	$M_3 = x_1 + \bar{x}_2 + \bar{x}_3$
4	1	0	0	$m_4 = x_1 \bar{x}_2 \bar{x}_3$	$M_4 = \bar{x}_1 + x_2 + x_3$
5	1	0	1	$m_5 = x_1 \bar{x}_2 x_3$	$M_5 = \bar{x}_1 + x_2 + \bar{x}_3$
6	1	1	0	$m_6 = x_1 x_2 \bar{x}_3$	$M_6 = \bar{x}_1 + \bar{x}_2 + x_3$
7	1	1	1	$m_7 = x_1 x_2 x_3$	$M_7 = \bar{x}_1 + \bar{x}_2 + \bar{x}_3$

Analise o circuito de quatro variáveis a seguir.



Considerando esse circuito, as funções f e g são, respectivamente,

- A** $\Sigma m(0,1,2,3,6,7,8,9)$ e $\Sigma m(2,3,6,7,10,14)$.
- B** $\Sigma m(4,5,10,11,12,13,14,15)$ e $\Sigma m(0,1,4,5,8,9,11,12,13,15)$.
- C** $\prod M(0,1,2,3,6,7,8,9)$ e $\prod M(0,1,4,5,8,9,11,12,13,15)$.
- D** $\prod M(4,5,10,11,12,13,14,15)$ e $\Sigma m(2,3,6,7,10,14)$.
- E** $\prod M(4,5,10,11,12,13,14,15)$ e $\prod M(2,3,6,7,10,14)$.

Exercício

Considere a seguinte tabela verdade, na qual estão definidas quatro entradas – A, B, C e D – e uma saída S.

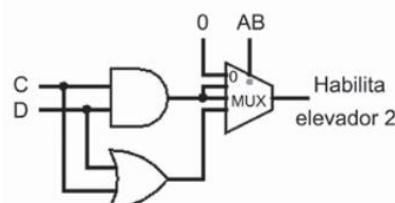
A	B	C	D	S
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

A menor expressão de chaveamento representada por uma soma de produtos correspondente à saída S é

- A AB'(D+C')+A'D'+ABC.
- B AD + A'BD'+A'BC+A'B'C'.
- C A'D' + AB'D+AB'C'+ABC.
- D (A'+D)(A+B+C')(A+B'+C+D').
- E (A+D')(A'+B'+C)(A'+B+C'+D).

Exercício

Um prédio de 4 andares, sendo o primeiro o andar térreo, é servido por 2 elevadores. Por motivo de economia de energia, o elevador 2 só é acionado se for solicitado em mais de 2 andares. Considere um circuito proposto para habilitar o acionamento do elevador 2 conforme é mostrado a seguir. Ele utiliza um multiplexador 4x1, cuja saída é selecionada através da composição dos sinais A e B, que indicam se os andares 1 e 2 solicitaram o serviço do elevador. Assim, o valor AB=10₍₂₎ indica que o primeiro andar solicitou elevador, mas não o segundo. Os sinais C e D indicam se os andares 3 e 4 solicitaram o serviço, respectivamente.



Com base na análise do circuito proposto para o problema, avalie as seguintes asserções e a relação proposta entre elas.

- I. O circuito não atende às especificações do projeto.

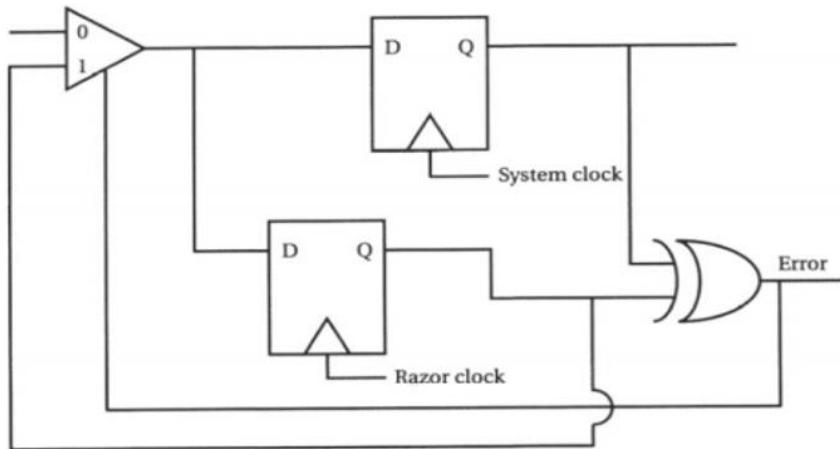
PORQUE

- II. A entrada superior do multiplexador com valor constante 0 indica que a saída será 0 independentemente dos valores dos sinais A, B, C e D.

A respeito dessas asserções, assinale a opção correta.

- A As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
- B As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.
- C A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- D A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- E As asserções I e II são proposições falsas.

O razor é uma arquitetura para desempenho better-than-worst-case que usa um registrador especializado, mostrado na figura, que mede e avalia os erros.



O registrador do sistema mantém o valor chaveado e é comandado por um *clock* de sistema *better-than-worst-case*. Um registrador adicional é comandado separadamente por um *clock* ligeiramente atrasado com relação ao do sistema. Se os resultados armazenados nos dois registradores são diferentes, então um erro ocorreu, provavelmente devido à temporização. A porta XOR detecta o erro e faz com que este valor seja substituído por aquele no registrador do sistema.

Wolf, W. High-performance embedded computing: architectures, applications, and methodologies. Morgan Kaufmann, 2007

Considerando essas informações, analise as afirmações a seguir.

- I. Sistemas digitais são tradicionalmente concebidos como sistemas assíncronos regidos por um *clock*.
- II. *Better-than-worst-case* é um estilo de projeto alternativo em que a lógica detecta e se recupera de erros, permitindo que o circuito possa operar com uma frequência maior.
- III. Nos sistemas digitais, o período de *clock* é determinado por uma análise cuidadosa para que os valores sejam armazenados corretamente nos registradores, com o período de *clock* alargado para abranger o atraso de pior caso.

É correto o que se afirma em

- A I, apenas.
- B III, apenas.
- C I e II, apenas.
- D II e III, apenas.
- E I, II e III.

Engenharia de Software

QUESTÃO 26

Um engenheiro de software planejou o desenvolvimento de um novo projeto, com prazo máximo de 220 dias, em seis fases: comunicação, planejamento, modelagem, construção, documentação e implantação. As fases seriam realizadas na sequência em que foram listadas. Exceção foi feita para as fases de construção e a documentação, que poderiam ocorrer em paralelo. Entretanto, a fase de implantação só poderia ocorrer se tanto construção quanto documentação estivessem encerradas.

A tabela a seguir apresenta a duração de cada fase do plano de desenvolvimento proposto.

Tabela - Fases e respectivas dependências e durações

Tabela - Fases e respectivas dependências e durações

#	Fase	Dependência	Duração (dias)
1	Comunicação	-	15
2	Planejamento	1	30
3	Modelagem	2	45
4	Construção	3	100
5	Documentação	3	40
6	Implantação	4,5	30

Gerência de Projetos (cont.)

Considerando o uso do Método do Caminho Crítico, e que o projeto tem prazo máximo de 220 dias, com início no dia 1, avalie as seguintes afirmações.

- I. A data mínima para o início da fase de implantação é o dia 191.
- II. O projeto possui um caminho crítico, que é 1-2-3-5-6.
- III. A folga livre da atividade documentação é de 60 dias.

É correto apenas o que se afirma em:

- A I.
- B I e II.
- C I e III.
- D II e III.
- E I, II e III.

QUESTÃO 19

Uma equipe está realizando testes com base nos códigos-fonte de um sistema. Os testes envolvem a verificação de diversos componentes individualmente, bem como das interfaces entre os componentes.

No contexto apresentado, essa equipe está realizando testes em nível de

- A unidade.
- B aceitação.
- C sistema e aceitação.
- D integração e sistema.
- E unidade e integração.

QUESTÃO 17 -----

Considerando que o gerente de qualidade é o responsável por definir os meios necessários para se obter um produto com a qualidade desejada, bem como por estabelecer técnicas para aferir a qualidade do produto, avalie as afirmações a seguir.

- I. O uso de processos de desenvolvimento padronizados, sem adaptações, independente do tipo de *software* a ser desenvolvido, assegura que o produto terá a qualidade desejada.
- II. O controle de qualidade pode ser realizado por meio de revisões, incluindo inspeções de programas e de artefatos de projeto.
- III. Fatores de qualidade de *software* estão diretamente relacionados a um único atributo interno de *software*.

É correto o que se afirma em

- A II, apenas.
- B III, apenas.
- C I e II, apenas.
- D I e III, apenas.
- E I, II e III.

QUESTÃO 25 -----

A engenharia de *software* considera diversos aspectos para a garantia da qualidade. Os requisitos funcionais definem como um sistema deverá se comportar em relação as suas funcionalidades básicas, já os requisitos não funcionais avaliam outros aspectos do *software*.

São exemplos de requisitos não funcionais a serem considerados em um *software*:

- A segurança, desempenho, estresse e sistema.
- B usabilidade, segurança, aceitação e confiabilidade.
- C usabilidade, segurança, desempenho e confiabilidade.
- D segurança, aceitação, testabilidade e confidencialidade.
- E usabilidade, confidencialidade, aceitação e confiabilidade.

QUESTÃO 9

O levantamento de requisitos é uma etapa fundamental do projeto de sistemas. Dependendo da situação encontrada, uma ou mais técnicas podem ser utilizadas para a elicitação dos requisitos. A respeito dessas técnicas, analise as afirmações a seguir.

- I. Workshop de requisitos consiste na realização de reuniões estruturadas e delimitadas entre os analistas de requisitos do projeto e representantes do cliente.
- II. Cenário consiste na observação das ações do funcionário na realização de uma determinada tarefa, para verificar os passos necessários para sua conclusão.
- III. As entrevistas são realizadas com os *stakeholders* e podem ser abertas ou fechadas.
- IV. A prototipagem é uma versão inicial do sistema, baseado em requisitos levantados em outros sistemas da organização.

É correto apenas o que se afirma em

- A I e II.
- B I e III.
- C II e IV.
- D I, III e IV.
- E II, III e IV.

QUESTÃO 10

Várias técnicas relacionadas à programação extrema (XP) são diretamente ligadas ao código, incluindo a refatoração, programação em pares e integração contínua. A programação em pares é a prática preferida dos desenvolvedores XP trabalhando em pares em um computador.

SCOTT, K. *O Processo Unificado Explicado*. Porto Alegre: Artmed, 2002.
BECK, K. *Programação Extrema (XP) Explicada*. Porto Alegre: Artmed, 2000.

A programação em pares auxilia no desenvolvimento de código de melhor qualidade quando os pares

- A elaboram e utilizam padrões de codificação conjuntamente, os quais, quando utilizados corretamente e apropriadamente, reduzem problemas individuais.
- B estão acostumados ao desenvolvimento e à propriedade coletiva, limitando-se a fazer pequenas mudanças na ocorrência de erros em tempo de execução.
- C minimizam os riscos de insucesso no projeto através da utilização de ferramentas para a geração automática de testes funcionais e protótipos de interface.
- D escrevem testes em separado e discutem os resultados posteriormente, o que dá a eles a chance de se sintonizarem antes de começarem a implementação.
- E trabalham em projetos complexos onde a codificação seja desenvolvida de forma conjunta, minimizando erros e agregando valor onde quer que o sistema necessite.

(Tem mais exercícios mas são só mais uns 500, sem tempo irmão)

Estrutura de Dados

QUESTÃO 16



Uma pilha é uma estrutura de dados que armazena uma coleção de itens de dados relacionados e que garante o seguinte funcionamento: o último elemento a ser inserido é o primeiro a ser removido. É comum na literatura utilizar os nomes *push* e *pop* para as operações de inserção e remoção de um elemento em uma pilha, respectivamente. O seguinte trecho de código em linguagem C define uma estrutura de dados pilha utilizando um vetor de inteiros, bem como algumas funções para sua manipulação.

```
#include <stdlib.h>
#include <stdio.h>
typedef struct {
    int elementos[100];
    int topo;
}pilha;
```

```
pilha * cria_pilha() {
    pilha * p = malloc(sizeof(pilha));
    p->topo = -1;
    return pilha;
}

void push(pilha *p, int elemento) {
    if (p->topo >= 99)
        return;
    p->elementos[++p->topo] = elemento;
}

int pop(pilha *p) {
    int a = p->elementos[p->topo];
    p->topo--;
    return a;
}
```

O programa a seguir utiliza uma pilha.

```
int main() {
    pilha * p = cria_pilha();
    push(p, 2);
    push(p, 3);
    push(p, 4);
    pop(p);
    push(p, 2);
    int a = pop(p) + pop(p);
    push(p, a);
    a += pop(p);
    printf("%d", a);
    return 0;
}
```

A esse respeito, avalie as afirmações a seguir.

- I. A complexidade computacional de ambas funções *push* e *pop* é $O(1)$.
- II. O valor exibido pelo programa seria o mesmo caso a instrução `a += pop(p);` fosse trocada por `a += a;`
- III. Em relação ao vazamento de memória (*memory leak*), é opcional chamar a função `free(p)`, pois o vetor usado pela pilha é alocado estaticamente.

É correto o que se afirma em

- A I, apenas.
- B III, apenas.
- C I e II, apenas.
- D II e III, apenas.
- E I, II e III.

QUESTÃO 29

A pilha é uma estrutura de dados que permite a inserção/remoção de itens dinamicamente seguindo a norma de último a entrar, primeiro a sair. Suponha que para uma estrutura de dados, tipo pilha, são definidos os comandos:

- PUSH (p, n): Empilha um número "n" em uma estrutura de dados do tipo pilha "p";
- POP (p): Desempilha o elemento no topo da pilha.

Considere que, em uma estrutura de dados tipo pilha "p", inicialmente vazia, sejam executados os seguintes comandos:

PUSH (p, 10)
PUSH (p, 5)
PUSH (p, 3)
PUSH (p, 40)
POP (p)
PUSH (p, 11)
PUSH (p, 4)
PUSH (p, 7)
POP (p)
POP (p)

Após a execução dos comandos, o elemento no topo da pilha "p" e a soma dos elementos armazenados na pilha p são, respectivamente,

- A 11 e 29.
 B 11 e 80.
 C 4 e 80.
 D 7 e 29.
 E 7 e 40.

QUESTÃO 21

No desenvolvimento de um software que analisa bases de DNA, representadas pelas letras A, C, G, T, utilizou-se as estruturas de dados: pilha e fila. Considere que, se uma sequência representa uma pilha, o topo é o elemento mais à esquerda; e se uma sequência representa uma fila, a sua frente é o elemento mais à esquerda.

Analise o seguinte cenário: "a sequência inicial ficou armazenada na primeira estrutura de dados na seguinte ordem: (A,G,T,C,A,G,T,T). Cada elemento foi retirado da primeira estrutura de dados e inserido na segunda estrutura de dados, e a sequência ficou armazenada na seguinte ordem: (T,T,G,A,C,T,G,A). Finalmente, cada elemento foi retirado da segunda estrutura de dados e inserido na terceira estrutura de dados e a sequência ficou armazenada na seguinte ordem: (T,T,G,A,C,T,G,A)".

Qual a única sequência de estruturas de dados apresentadas a seguir pode ter sido usada no cenário descrito acima?

- A Fila - Pilha - Fila.
 B Fila - Fila - Pilha.
 C Fila - Pilha - Pilha.
 D Pilha - Fila - Pilha.
 E Pilha - Pilha - Pilha.

QUESTÃO 14

Um programador propôs um algoritmo não-recursivo para o percurso em preordem de uma árvore binária com as seguintes características.

- Cada nó da árvore binária é representado por um registro com três campos: `chave`, que armazena seu identificador; `esq` e `dir`, ponteiros para os filhos esquerdo e direito, respectivamente.
 - O algoritmo deve ser invocado inicialmente tomando o ponteiro para o nó raiz da árvore binária como argumento.
 - O algoritmo utiliza `push()` e `pop()` como funções auxiliares de empilhamento e desempilhamento de ponteiros para nós de árvore binária, respectivamente.
-

A seguir, está apresentado o algoritmo proposto, em que λ representa o ponteiro nulo.

```
Procedimento preordem (ptraiz : PtrNoArvBin)
    Var ptr : PtrNoArvBin;
    ptr := ptraiz;
    Enquanto (ptr ≠ λ) Faça
        escreva (ptr;.chave);
        Se (ptr;.dir ≠ λ) Então
            push(ptr;.dir);
        Se (ptr;.esq ≠ λ) Então
            push(ptr;.esq);
        ptr := pop();
    Fim_Enquanto
Fim_Procedimento
```

Com base nessas informações e supondo que a raiz de uma árvore binária com n nós seja passada ao procedimento `preordem()`, julgue os itens seguintes.

Com base nessas informações e supondo que a raiz de uma árvore binária com n nós seja passada ao procedimento `preordem()`, julgue os itens seguintes.

- I O algoritmo visita cada nó da árvore binária exatamente uma vez ao longo do percurso.
- II O algoritmo só funcionará corretamente se o procedimento `pop()` for projetado de forma a retornar λ caso a pilha esteja vazia.
- III Empilhar e desempilhar ponteiros para nós da árvore são operações que podem ser implementadas com custo constante.
- IV A complexidade do pior caso para o procedimento `preordem()` é $O(n)$.

Assinale a opção correta.

- (A) Apenas um item está certo.
- (B) Apenas os itens I e IV estão certos.
- (C) Apenas os itens I, II e III estão certos.
- (D) Apenas os itens II, III e IV estão certos.
- (E) Todos os itens estão certos.

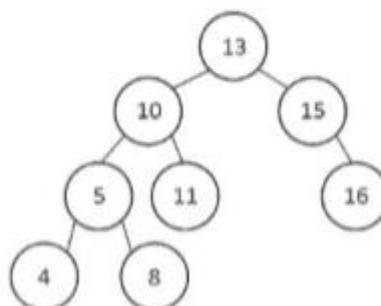
Questão 09

Assunto: estruturas de dados. [\[Índice\]](#)

Uma árvore AVL é um tipo de árvore binária balanceada na qual a diferença entre as alturas de suas subárvore da esquerda e da direita não pode ser maior do que 1 para qualquer nó. Após a inserção de um novo nó em uma AVL, a raiz da subárvore de nível mais baixo no qual o novo nó foi inserido é marcada. Se a altura de seus filhos diferir em mais de uma unidade, é realizada uma rotação simples ou uma rotação dupla para igualar suas alturas.

LAFORE, R. *Data Structures & algorithms in Java*. Indianópolis: Sams Publishing, 2003 (adaptado).

A seguir, é apresentado um exemplo de árvore AVL.



Pelo exposto no texto acima, após a inserção de um novo nó com valor 3 na árvore AVL exemplificada, é correto afirmar que ela ficará com a seguinte configuração

ENADE 2017

Alternativa A: Árvore com raiz 13, filho esquerdo 5 (filhos 4 e 8) e filho direito 15 (filho 16). O nó 3 está circulado em azul.

Alternativa B: Árvore com raiz 13, filho esquerdo 10 (filhos 4, 5 e 8) e filho direito 15 (filhos 11 e 16). O nó 3 está circulado em azul.

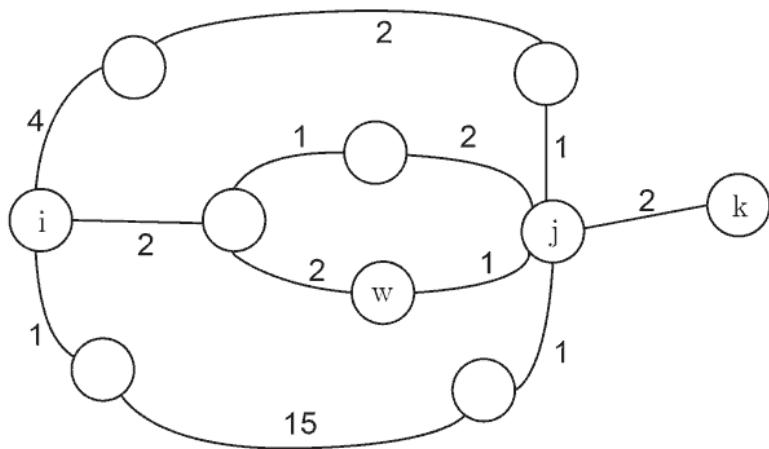
Alternativa C: Árvore com raiz 10, filho esquerdo 5 (filhos 4 e 11) e filho direito 13 (filhos 15 e 16). O nó 3 está circulado em azul.

Alternativa D: Árvore com raiz 10, filho esquerdo 5 (filhos 4 e 8) e filho direito 13 (filhos 11 e 15). O nó 3 está circulado em azul.

Alternativa E: Árvore com raiz 10, filho esquerdo 4 (filhos 3 e 5) e filho direito 13 (filhos 11 e 16).

Grafos

A figura a seguir exibe um **grafo** que representa um mapa rodoviário, no qual os vértices representam cidades e as arestas representam vias. Os pesos indicam o tempo atual de deslocamento entre duas cidades.



Grafo da questão 24

Considerando que os tempos de ida e volta são iguais para qualquer via, avalie as afirmações a seguir acerca desse grafo.

- Dado o vértice de origem i , o algoritmo de Dijkstra encontra o menor tempo de deslocamento entre a cidade i e todas as demais cidades do grafo.
- Uma árvore geradora de custo mínimo gerada pelo algoritmo de Kruskal contém um caminho de custo mínimo cuja origem é i e cujo destino é k .
- Se um caminho de custo mínimo entre os vértices i e k contém o vértice w , então o subcaminho de origem w e destino k deve também ser mínimo.

É correto o que se afirma em

- (A) I, apenas.
- (B) II, apenas.
- (C) I e III, apenas.
- (D) II e III, apenas.
- (E) I, II e III.

Linguagens Formais

Questão 64 – ENADE 2005

Considere a necessidade de se implementar um componente de *software* que realiza cálculos de expressões matemáticas simples para as operações básicas (soma, subtração, multiplicação, divisão e exponenciação). O *software* reproduz na tela do computador a entrada, os resultados parciais e o resultado final da expressão e, ainda, trata os operadores de exponenciação, multiplicação e divisão com precedência sobre os operadores de soma e subtração. Para obter o referido *software*, é correto que o projetista

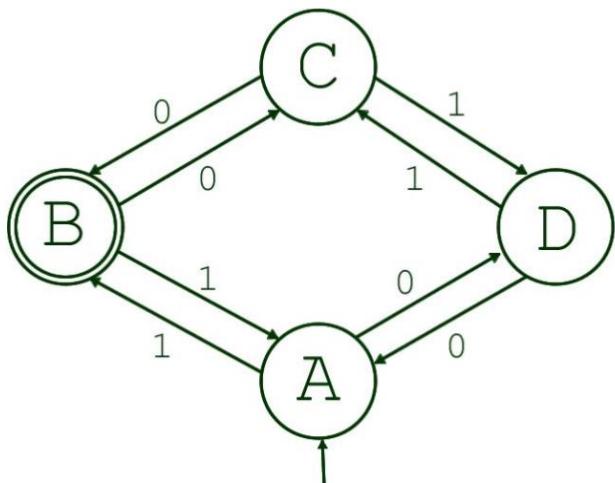
- defina uma cadeia de caracteres para armazenar e imprimir toda a expressão de entrada.
- defina uma gramática regular para identificar as expressões aritméticas válidas.
- defina um reconhecedor de linguagem regular com autômato finito determinístico.
- especifique a ordem de precedência dos operadores com uma notação de gramática livre de contexto.

Estão certos apenas os itens

- (A) I e II.
- (B) III e IV.
- (C) I, II e IV.
- (D) I, III e IV,
- (E) II, III e IV.

Questão 80 - ENADE 2005

Que cadeia é reconhecida pelo autômato representado pelo diagrama de estados ao lado?



- (A) 101010,
- (B)** 111011000,
- (C) 11111000,
- (D) 10100,
- (E) 00110011,

Questão 15 - ENADE 2014 - CCO

Considere as seguintes expressões regulares cujo alfabeto é {a, b}.

$$R1 = a(a \cup b)^*$$

$$R2 = b(a \cup b)^*$$

Se $L(R)$ é a linguagem associada a uma expressão regular R , é correto afirmar que

- (A) $L(R1) = L(R2)$.
- (B)** $L(R2) = \{w \mid w \text{ termina com } b\}$.
- (C)** existe um autômato finito determinístico cuja linguagem é igual a $L(R1) \cup L(R2)$.
- (D) se $R3$ é uma expressão regular tal que $L(R3) = L(R1) \cap L(R2)$, então $L(R3)$ é uma linguagem infinita.
- (E) um autômato finito não determinístico que reconheça $L(R1) \cup L(R2)$ tem, pelo menos, quatro estados.

Questão 29 - ENADE 2014 - ECO

Expressões regulares constituem formas sucintas de descrever linguagens regulares. Uma de suas aplicações é descrever padrões a serem procurados em um texto. As expressões regulares $R1$, $R2$, $R3$ e $R4$ a seguir utilizam a seguinte convenção: o fecho de Kleene é denotado por $*$ e a união é denotada pelo símbolo $|$.

$$\begin{aligned} R1 &= a^*ba^*ba^* \\ R2 &= a^*(a \mid b)a(a \mid b)^* \\ R3 &= a^*ab^*a(a \mid b) \\ R4 &= (a \mid b)^* \end{aligned}$$

Em relação às linguagens definidas pelas expressões regulares apresentadas, conclui-se que a cadeia **abbb** está contida apenas nas linguagens definidas por

- (A)** $R1$ e $R4$,
- (B) $R2$ e $R3$.
- (C) $R2$ e $R4$.
- (D) $R1$ e $R3$.
- (E) $R2$, $R3$ e $R4$.

Matemática Discreta

Seja o universo $U=\{10,20,30,40\}$ e o conjunto dos números naturais N . com base no conhecimento sobre lógica de predicados, avalie as afirmações a seguir.

- I. $H = (\forall x \in N)(\exists y \in U)(x < y)$ é válida.
- II. $H = (\forall x \in N)(\exists y \in N)(y < x)$ é válida.
- III. $H = (\forall x \in U)(\exists y \in U)(x > y)$ é inválida, sendo $x=10$ um contra-exemplo.

É correto o que se afirma em [Clique para adicionar texto](#)

- A. I, apenas
- B. III, apenas
- C. I e II, apenas
- D. II e III, apenas
- E. I, II e III

Considere o seguinte argumento:

- 1 – Se existe fogo, então existe oxigênio.
- 2 – Não há oxigênio.
- 3 – Então não há fogo.

A regra de inferência que justifica a validade do argumento acima é

- A. $\frac{P \rightarrow Q, \neg P}{\neg Q}$
- B. $\frac{P \rightarrow Q, \neg Q}{\neg P}$
- C. $\frac{P \rightarrow Q, Q}{P}$
- D. $\frac{P \rightarrow Q, \neg Q}{\neg \neg P}$
- E. $\frac{P \rightarrow Q, P}{Q}$

Considere as proposições lógicas simples P,Q,R:
 P: o programador lê a literatura técnica
 Q: o programador conhece o idioma inglês
 R: o programador será selecionado
 Pretende-se demonstrar a validade ou invalidade do seguinte argumento:
 Se o programador lê a literatura técnica, então ele conhece inglês.
 Se o programador conhece o idioma inglês, então ele será selecionado.
 O programador não será selecionado ou ele lê a literatura técnica.
 Logo, o programador lê a literatura técnica se e somente se conhece o idioma inglês.
 Considerando as colocações acima, avalie as afirmações a seguir.

- I. As premissas do argumento podem ser expressas na forma: $P \rightarrow Q$, $Q \rightarrow R$ e $\neg R \vee P$. A conclusão do argumento pode ser expressa na forma: $P \leftrightarrow Q$.
- II. A validade do argumento se demonstra com os passos: $\neg Q \vee P$ (equivalente de uma premissa), $P \rightarrow R$ (transitividade da implicação a partir das premissas) e conclusão $Q \leftrightarrow R$ (conjunção de duas proposições condicionais e transformação em bicondicional).
- III. A validade do argumento se demonstra com os passos: $R \rightarrow P$ (equivalente de uma premissa), $Q \rightarrow P$ (transitividade da implicação), chegamos à conclusão $Q \leftrightarrow R$ (conjunção de duas proposições condicionais e transformação em bicondicional).
- IV. As premissas do argumento podem ser expressas na forma: $P \rightarrow Q$, $Q \rightarrow R$ e $\neg R \rightarrow P$ e a conclusão do argumento acima pode ser expressa na forma: $P \rightarrow Q$.
- V. A invalidade do argumento acima se demonstra desta forma: a proposição lógica $P \leftrightarrow Q$ é diferente das premissas $P \rightarrow Q$, $Q \rightarrow R$ e $\neg R \vee P$.

É correto apenas o que se afirma em

- A. I e III
 B. II e IV
 C. I, III e V
 D. I, II, IV e V
 E. II, III, IV e V

Programação C

Um país utiliza moedas de 1, 5, 10, 25 e 50 centavos. Um programador desenvolveu o método a seguir, que implementa a estratégia gulosa para o problema do troco mínimo. Esse método recebe como parâmetro um valor inteiro, em centavos, e retorna um array no qual cada posição indica a quantidade de moedas de cada valor.

```
public static int[] troco(int valor){
    int[] moedas = new int[5];

    moedas[4] = valor / 50;
    valor = valor % 50;
    moedas[3] = valor / 25;
    valor = valor % 25;
    moedas[2] = valor / 10;
    valor = valor % 10;
    moedas[1] = valor / 5;
    valor = valor % 5;
    moedas[0] = valor;
    return moedas;
}
```

Considerando o método apresentado, avalie as asserções a seguir e a relação proposta entre elas.

I. O método guloso encontra o menor número de moedas para o valor de entrada, considerando as moedas do país.

PORQUE

II. Métodos gulosos sempre encontram a solução ótima global.

A respeito dessas asserções, assinale a opção correta.

- (A) As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
 (B) As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.
 (C) A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
 (D) A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
 (E) As asserções I e II são proposições falsas.

ENADE CCO - 2017

Q. 25 A sequência de Fibonacci é uma sequência de números inteiros que começa em 1, a que se segue 1, e na qual cada elemento subsequente é a soma dos dois anteriores. A função fib a seguir calcula o n-ésimo elemento da sequência de Fibonacci.

Considerando a implementação cima, avalia as afirmações a seguir.

- I. A complexidade de tempo da função fib é exponencial no valor de n.
- II. A complexidade de espaço da função fib é exponencial no valor de n.
- III. É possível implementar uma versão iterativa da função fib com complexidade de tempo linear no valor de n e complexidade de espaço constante.

```
unsigned int fib ( unsigned int n ) {
    if ( n < 2 ) return 1;
    return fib( n - 2 ) + fib ( n - 1 );
}
```

- A) I, apenas.
 B) II, apenas.
 C) I e III, apenas.
 D) II e III, apenas.
 E) I, II e III.

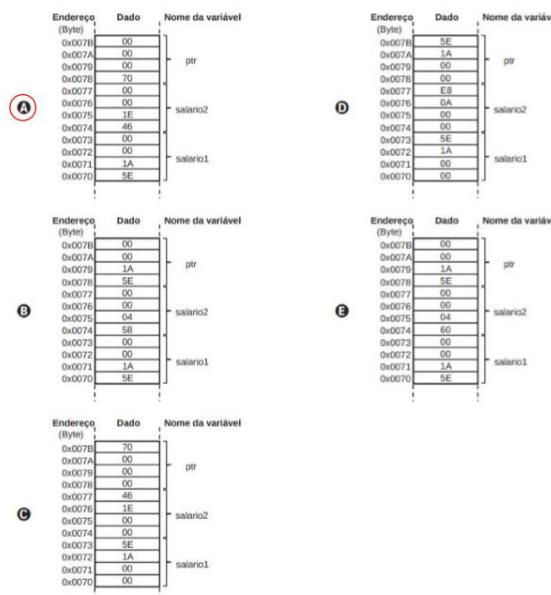
ENADE ECO - 2017

Q. 22 Considere o código em C, a seguir, compilado em um sistema computacional de 32bits, que exemplifica o uso do tipo de dados ponteiro.

```
#include <stdio.h>

int main ( void ) {
    int salario1, salario2;
    int *ptr;
    salario1 = 6750;
    ptr = &salario1;
    salario2 = *ptr + 1000;
    return 0;
}
```

A partir das informações apresentadas, assinale a opção que apresenta a memória principal e seu conteúdo, após o programa ser executado.



ENADE CCO - 2017

Q. 18 O algoritmo a seguir recebe um vetor v de números inteiros e rearranja esse vetor de tal forma que seus elementos, ao final, estejam ordenados de forma crescente. Considerando que nesse algoritmo há erros de lógica que devem ser corrigidos para que os elementos sejam ordenados de forma crescente, assinale a opção correta no que se refere às correções adequadas.

A) A linha 04 deve ser corrigida da seguinte forma:
for (i = 1; i < n - 1; i++) e a linha 13, do seguinte modo: **v[j - 1] = chave;** B)

A linha 04 deve ser corrigida da seguinte forma: **for (i = 1; i < n - 1; i++)** e a linha 07, do seguinte modo **j = i + 1;**

C) A linha 07 deve ser corrigida da seguinte forma: **j = i + 1;** e a linha 08, do seguinte modo **while(j >= 0 && v[j] > chave);**

```
1 void ordena ( int *v int n )
2 {
3     int i, j, chave;
4     for( i = 0; i < n; i++ )
5     {
6         chave = v[i];
7         j = i - 1;
8         while( j >= 0 && v[j] < chave )
9         {
10            v[j-1] = v[j];
11            j = j - 1;
12        }
13        v[j+1] = chave;
14    }
15 }
```

D) A linha 08 deve ser corrigida da seguinte forma:
while(j >= 0 && v[j] > chave) e a linha 10, do seguinte modo: **v[j + 1] = v[j];**

E) A linha 10 deve ser corrigida da seguinte forma: **v[j + 1] = v[j];** e a linha 13 do seguinte modo: **v[j - 1] = chave;**

Sistemas distribuídos

QUESTÃO 29

Uma antiga empresa de desenvolvimento de software resolveu atualizar toda sua infraestrutura computacional adquirindo um sistema operacional multitarefa, processadores *multi-core* (múltiplos núcleos) e o uso de uma linguagem de programação com suporte a *threads*.

O sistema operacional multitarefa de um computador é capaz de executar vários processos (programas) em paralelo. Considerando esses processos implementados com mais de uma *thread* (*multi-threads*), analise as afirmações abaixo.

- I. Os ciclos de vida de processos e *threads* são idênticos.
- II. *Threads* de diferentes processos compartilham memória.
- III. Somente processadores *multi-core* são capazes de executar programas *multi-threads*.
- IV. Em sistemas operacionais multitarefa, *threads* podem migrar de um processo para outro.

É correto apenas o que se afirma em

- A I.
- B II.
- C I e III.
- D I e IV.
- E II e IV.

QUESTÃO 15

Suponha que seja necessário desenvolver uma ferramenta que apresente o endereço IP dos múltiplos roteadores, salto a salto, que compõem o caminho do hospedeiro em que a ferramenta é executada até um determinado destino (segundo seu endereço IP), assim como o *round-trip time* até cada roteador. Tal ferramenta precisa funcionar na Internet atual, sem demandar mudanças em roteadores nem a introdução de novos protocolos.

Considerando o problema acima, qual dos seguintes protocolos representaria a melhor (mais simples e eficiente) solução?

- A IP: Internet Protocol.
- B UDP: User Datagram Protocol.
- C TCP: Transmission Control Protocol.
- D ICMP: Internet Control Message Protocol.
- E DHCP: Dynamic Host Configuration Protocol.

QUESTÃO 16

Um navegador Web executa em um hospedeiro A, em uma rede de uma organização, e acessa uma página localizada de um servidor Web em um hospedeiro B, situado em outra rede na Internet. A rede em que A se situa conta com um servidor DNS local. Um profissional deseja fazer uma lista com a sequência de protocolos empregados e comparar com o resultado apresentado por uma ferramenta de monitoramento executada no hospedeiro A. A lista assume que

- i) todas as tabelas com informações temporárias e caches estão vazias;

- ii) o hospedeiro cliente está configurado com o endereço IP do servidor DNS local.

Qual das sequências a seguir representa a ordem em que mensagens, segmentos e pacotes serão observados em um meio físico ao serem enviados pelo hospedeiro A?

- A ARP, DNS/UDP/IP, TCP/IP e HTTP/TCP/IP.
- B ARP, DNS/UDP/IP, HTTP/TCP/IP e TCP/IP.
- C DNS/UDP/IP, ARP, HTTP/TCP/IP e TCP/IP.
- D DNS/UDP/IP, ARP, TCP/IP e HTTP/TCP/IP.
- E HTTP/TCP/IP, TCP/IP, DNS/UDP/IP e ARP.

QUESTÃO 25

Um provedor de serviços de segurança de redes e sistemas distribuídos enumerou três componentes de rede essenciais para a garantia da segurança dos dados corporativos: *firewall* de rede; sistemas de prevenção e detecção de intrusão; e gateways antivírus.

Acerca desses componentes de rede, assinale a opção correta.

- A** Os gateways antivírus trabalham no nível da camada de rede e verificam o fluxo de dados em busca de assinaturas de vírus conhecidas.

B O *firewall* de rede deve ser configurado para detectar transferência de informação através de um canal camouflado (*covert channel*) baseado em túneis.

C Um *firewall* de camada de rede (*network layer firewall*) permite uma filtragem mais detalhada dos dados que um *firewall* de camada de aplicação (*application layer firewall*) ao custo de um pior desempenho.

D Os sistemas de prevenção de intrusão são vistos como uma extensão do *firewall* e são capazes de detectar anomalias de tráfego ou conteúdo malicioso antes que eles alcancem a rede.

E O sistema de detecção de intrusão é capaz de identificar ataques iniciados dentro da rede protegida e agir proativamente para neutralizar a ameaça.

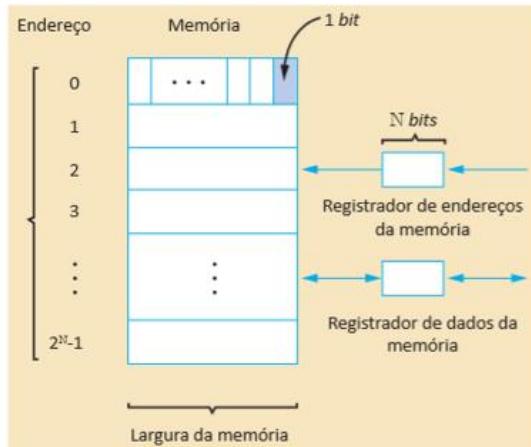
Sistemas operacionais

QUESTÃO 12

Em um computador, a memória é a unidade funcional que armazena e recupera operações e dados. Tipicamente, a memória de um computador usa uma técnica chamada acesso aleatório, que permite o acesso a qualquer uma de suas posições (células). As memórias de acesso aleatório são divididas em células de tamanho fixo, estando cada célula associada a um identificador numérico único chamado endereço. Todos os acessos à memória referem-se a um endereço específico e deve-se sempre buscar ou armazenar o conteúdo completo de uma célula, ou seja, a célula é a unidade mínima de acesso.

SCHNEIDER, G. M.; GERSTING, J. L. *An invitation to computer science*. 6. ed. Boston: MA: Course Technology, Cengage Learning, 2009 (adaptado).

A figura que se segue apresenta a estrutura de uma unidade de memória de acesso aleatório.



Considerando o funcionamento de uma memória de acesso aleatório, avalie as afirmações a seguir.

- Se a largura do registrador de endereços da memória for de 8 bits, o tamanho máximo dessa unidade de memória será de 256 células.
- Se o registrador de dados da memória tiver 8 bits, será necessária mais de uma operação para armazenar o valor inteiro 2 024 nessa unidade de memória.
- Se o registrador de dados da memória tiver 12 bits, é possível que a largura da memória seja de 8 bits.

É correto o que se afirma em

- A** I, apenas.
B III, apenas.
C I e II, apenas.
D II e III apenas.
E I, II e III.

QUESTÃO 29

O projetista do gerenciador de memória de um novo sistema operacional precisa escolher entre os algoritmos de substituição de páginas FIFO (*First In First Out* - o primeiro a entrar é o primeiro a sair) e LRU (*Least Recently Used* - menos recentemente usado). Para isso, avaliou o número de faltas de página obtidas em ambos os algoritmos para o tamanho de memória de 4 páginas, utilizando a sequência de acessos às páginas 1-2-3-4-1-2-5-1-2-3-4-5 de um processo e memória inicialmente vazia.

Com base nessa simulação, avalie as asserções a seguir e a relação proposta entre elas.

- I. Na simulação proposta, é possível observar que os algoritmos FIFO e LRU apresentam o mesmo desempenho.

PORQUE

- II. Os parâmetros utilizados na simulação são insuficientes para determinar a diferença de funcionamento entre os algoritmos.

A respeito dessas asserções, assinale a opção correta.

- A** As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
- B** As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.
- C** A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- D** A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- E** As asserções I e II são proposições falsas.



QUESTÃO 31

Na transmissão de dados em sistemas computacionais, o dispositivo de verificação de erros conhecido como bit de paridade consiste da adição de um bit extra durante a transmissão. O valor associado a esse bit é uma função da quantidade de bits de dados iguais a 1 a serem transmitidos.

Nesse contexto, considere a transmissão de 7 bits de dados, com um bit extra de paridade, em um sistema de comunicação no qual a probabilidade de transmitir um bit de forma incorreta é igual a 10^{-6} e independe de outros erros ocorridos. O bit de paridade também pode sofrer erros.

A probabilidade da ocorrência de transmissão de 2 bits errados, que seria erroneamente detectada como uma transmissão sem erros, é

- A** $1,0 \times 10^{-12}$.
- B** $2,0 \times 10^{-12}$.
- C** $2,8 \times 10^{-11}$.
- D** $2,0 \times 10^{-6}$.
- E** $2,8 \times 10^{-5}$.



QUESTÃO 19

Uma alternativa para o aumento de desempenho de sistemas computacionais é o uso de processadores com múltiplos núcleos, chamados *multicores*. Nesses sistemas, cada núcleo, normalmente, tem as funcionalidades completas de um processador, já sendo comuns, atualmente, configurações com 2, 4 ou mais núcleos. Com relação ao uso de processadores *multicores*, e sabendo que *threads* são estruturas de execução associadas a um processo, que compartilham suas áreas de código e dados, mas mantêm contextos independentes, analise as seguintes asserções.

Ao dividirem suas atividades em múltiplas *threads* que podem ser executadas paralelamente, aplicações podem se beneficiar mais efetivamente dos diversos núcleos dos processadores *multicores*
porque

o sistema operacional nos processadores *multicores* pode alocar os núcleos existentes para executar simultaneamente diversas seqüências de código, sobrepondo suas execuções e, normalmente, reduzindo o tempo de resposta das aplicações às quais estão associadas.

Acerca dessas asserções, assinale a opção correta.

- A As duas asserções são proposições verdadeiras, e a segunda é uma justificativa correta da primeira.
- B As duas asserções são proposições verdadeiras, mas a segunda não é uma justificativa correta da primeira.
- C A primeira asserção é uma proposição verdadeira, e a segunda, uma proposição falsa.
- D A primeira asserção é uma proposição falsa, e a segunda, uma proposição verdadeira.
- E Tanto a primeira quanto a segunda asserções são proposições falsas.

QUESTÃO 33

Compiladores de linguagens de programação traduzem programas-fonte, em uma linguagem de entrada, para programas-objeto, em uma linguagem de saída. Durante o processo de tradução, o compilador deve verificar se as sentenças do programa-fonte estão sintaticamente corretas. Esse processo de análise sintática pode ser realizado construindo-se uma árvore de análise segundo duas principais abordagens: *top-down*, quando a árvore é investigada da raiz às folhas; ou *bottom-up*, das folhas à raiz. Acerca desse assunto, julgue os itens seguintes.

- I A análise *top-down* é adequada quando a linguagem de entrada é definida por uma gramática recursiva à esquerda.
- II Independentemente da abordagem adotada, *top-down* ou *bottom-up*, o analisador sintático utiliza informações resultantes da análise léxica.
- III Se os programas em uma linguagem podem ser analisados tanto em abordagem *top-down* como em *bottom-up*, a gramática dessa linguagem é ambígua.
- IV A análise *bottom-up* utiliza ações comumente conhecidas como deslocamentos e reduções sobre as sentenças do programa-fonte.

Estão certos apenas os itens

- A I e II.
- B I e III.
- C II e IV.
- D I, III e IV.
- E II, III e IV.

QUESTÃO 71

Sistemas operacionais de tempo real são utilizados em controle de processos automatizados, em que o tempo de resposta a determinados eventos é um fator crítico. Com relação a esse assunto, julgue os itens seguintes.

- I Sistemas de tempo real estritos (*hard real-time*) não utilizam dispositivos de memória secundária (como discos), pois estes não oferecem garantia de término das operações dentro de uma quantidade máxima de tempo.
- II Um sistema operacional de propósito geral pode ser modificado para ser de tempo real atribuindo-se prioridades fixas para cada um dos processos.
- III O escalonamento mais utilizado por sistemas operacionais de tempo real é o *shortest-job-first* (tarefa mais curta primeiro).

Assinale a opção correta.

- **A** Apenas um item está certo.
B Apenas os itens I e II estão certos.
C Apenas os itens I e III estão certos.
D Apenas os itens II e III estão certos.
E Todos os itens estão certos.

QUESTÃO 29

Uma antiga empresa de desenvolvimento de *software* resolveu atualizar toda sua infraestrutura computacional adquirindo um sistema operacional multitarefa, processadores *multi-core* (múltiplos núcleos) e o uso de uma linguagem de programação com suporte a *threads*.

O sistema operacional multitarefa de um computador é capaz de executar vários processos (programas) em paralelo. Considerando esses processos implementados com mais de uma *thread* (*multi-threads*), analise as afirmações abaixo.

- I. Os ciclos de vida de processos e *threads* são idênticos.
- II. *Threads* de diferentes processos compartilham memória.
- III. Somente processadores *multi-core* são capazes de executar programas *multi-threads*.
- IV. Em sistemas operacionais multitarefa, *threads* podem migrar de um processo para outro.

É correto apenas o que se afirma em

- **A** I.
B II.
C I e III.
D I e IV.
E II e IV.

QUESTÃO 28**ITEM 2489**

A virtualização permite que um único computador hospede múltiplas máquinas virtuais, cada uma com seu próprio sistema operacional. Essa técnica tem ganhado importância nos dias atuais e vem sendo utilizada para resolver diversos tipos de problemas.

Considerando os diversos aspectos a serem considerados na utilização da virtualização, avalie as afirmações abaixo.

- I. Um sistema operacional sendo executado em uma máquina virtual utiliza um subconjunto da memória disponível na máquina real.
- II. Uma das aplicações da virtualização é a disponibilização de múltiplos sistemas operacionais para teste de software.
- III. A virtualização só pode ser utilizada em sistemas operacionais *Linux*.
- IV. Um sistema operacional executado em uma máquina virtual apresenta um desempenho superior ao que alcançaria quando executado diretamente na mesma máquina real.

É correto apenas o que se afirma em

- A** I.
→ III.
C I e II.
D II e IV.
E III e IV.