

Lista de Exercícios

01.

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.

03.

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.

02.

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.

04.

Sabendo que a principal tarefa de um sistema será de classificação em domínios complexos, um gerente de projetos precisa decidir como vai incorporar essa capacidade em um sistema computacional a fim de torná-lo inteligente. Existem diversas técnicas de inteligência computacional / artificial que possibilitam isso.

Nesse contexto, a técnica de inteligência artificial mais indicada para o gerente é

- A** lógica nebulosa.
- B** árvores de decisão.
- C** redes neurais artificiais.
- D** ACO (do inglês, *Ant-Colony Optimization*).
- E** PSO (do inglês, *Particle Swarm Optimization*).

05.

O problema **P versus NP** é um problema ainda não resolvido e um dos mais estudados em Computação. Em linhas gerais, deseja-se saber se todo problema cuja solução pode ser eficientemente verificada por um computador, também pode ser eficientemente obtida por um computador. Por "eficientemente" ou "eficiente" significa "em tempo polinomial".

A classe dos problemas cujas soluções podem ser eficientemente obtidas por um computador é chamada de **classe P**. Os algoritmos que solucionam os problemas dessa classe têm complexidade de pior caso polinomial no tamanho das suas entradas.

Para alguns problemas computacionais, não se conhece solução eficiente, isto é, não se conhece algoritmo eficiente para resolvê-los. No entanto, se para uma dada solução de um problema é possível verificá-la eficientemente, então o problema é dito estar em NP. Dessa forma, a classe de problemas para os quais suas soluções podem ser eficientemente verificadas é chamada de **classe NP**.

Um problema é dito ser **NP-completo** se pertence à classe NP e, além disso, se qualquer outro problema na classe NP pode ser eficientemente transformado nesse problema. Essa transformação eficiente envolve as entradas e saídas dos problemas.

Considerando as noções de complexidade computacional apresentadas acima, analise as afirmações que se seguem.

- I. Existem problemas na classe P que não estão na classe NP.
- II. Se o problema A pode ser eficientemente transformado no problema B e B está na classe P, então A está na classe P.
- III. Se $P = NP$, então um problema NP-completo pode ser solucionado eficientemente.
- IV. Se P é diferente de NP, então existem problemas na classe P que são NP-completos.

É correto apenas o que se afirma em

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

06.

Escopo dinâmico: para as linguagens com escopo dinâmico, a vinculação das variáveis ao escopo é realizada em tempo de execução. (...) Se uma variável é local ao bloco, então o uso da dada variável no bloco será sempre vinculado àquela local. Contudo, se a variável for não-local, a sua vinculação depende da ordem de execução, a última vinculada na execução. A consequência disso é que, em um mesmo bloco de comandos, um identificador pode ter significados diferentes, e o programador precisa ter a ideia precisa de qual variável está sendo usada.

de MELO, A. C. V.; da SILVA, F. S. C. **Princípios de Linguagens de Programação**. São Paulo: Edgard Blücher, 2003. p.65.

Suponha que uma linguagem de programação tenha sido projetada com vinculação e verificação estáticas para tipos de variáveis, além de passagem de parâmetros por valor. Também é exigido pela especificação da linguagem que programas sejam compilados integralmente e que não é permitido compilar bibliotecas separadamente. Durante uma revisão da especificação da linguagem, alguém propôs que seja adicionado um mecanismo para suporte a variáveis com escopo dinâmico.

A respeito da proposta de modificação da linguagem, analise as seguintes afirmações.

- I. As variáveis com escopo dinâmico podem ser tratadas como se fossem parâmetros para os subprogramas que as utilizam, sem que o programador tenha que especificá-las ou declarar seu tipo (o compilador fará isso). Assim, elimina-se a necessidade de polimorfismo e é possível verificar tipos em tempo de compilação.
- II. Como diferentes subprogramas podem declarar variáveis com o mesmo nome mas com tipos diferentes, se as variáveis com escopo dinâmico não forem declaradas no escopo onde são referenciadas, será necessário que a linguagem suporte polimorfismo de tipos.
- III. Se as variáveis dinâmicas forem declaradas tanto nos escopos onde são criadas como nos subprogramas em que são referenciadas, marcadas como tendo escopo dinâmico, será possível identificar todos os erros de tipo em tempo de compilação.

É correto apenas o que se afirma em

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