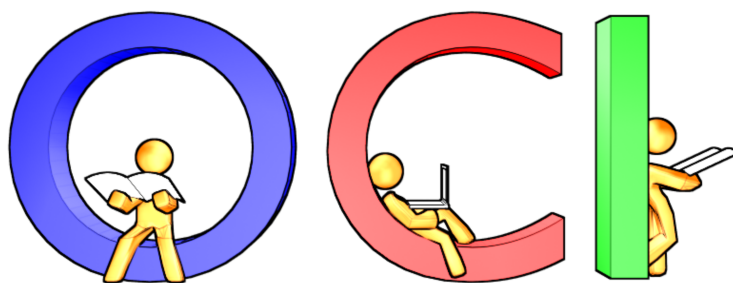


# IV Olimpíada Cearense de Informática



**20 de Setembro 2014**  
**Modalidade Programação**

**Leia atentamente as instruções:**

- Confira se os dados impressos no Cartão-Resposta correspondem aos seus. Caso haja alguma irregularidade, comunique-a imediatamente ao Aplicador da Prova.
- Não serão permitidos empréstimos de materiais, consultas e comunicação entre os candidatos, tampouco o uso de livros e apontamentos. Relógios e aparelhos eletrônicos em geral deverão ser desligados. O não cumprimento dessas exigências ocasionará a exclusão do candidato deste Exame.
- Aguarde o Aplicador da Prova autorizar a abertura do Caderno de Prova. Após a autorização, confira todas as questões antes de iniciar a Prova.
- Este Caderno de Prova contém 25 (vinte e cinco) questões objetivas, cada qual com apenas 1 (uma) alternativa correta. No Cartão-Resposta, preencha completamente, com caneta de tinta azul ou preta, o retângulo correspondente à alternativa que julgar correta para cada questão.
- No Cartão-Resposta, anulam a questão: a marcação de mais de uma alternativa em uma mesma questão, as rasuras e o preenchimento além dos limites do retângulo destinado para cada marcação. Não haverá substituição do Cartão-Resposta em nenhuma hipótese.
- Não serão permitidas perguntas ao Aplicador da Prova sobre as questões da Prova.
- A duração desta prova será de 3 (três) horas, já incluído o tempo para o preenchimento do Cartão-Resposta.
- O tempo mínimo para ausentar-se definitivamente da sala é de 1 (uma) hora.
- Ao concluir a prova, permaneça em seu lugar e comunique ao Aplicador da Prova, sinalizando com uma de suas mãos.
- Aguarde autorização para devolver, em separado, o Caderno de Prova e o Cartão-Resposta assinado.



**1.** Em um cesto há 15 lenços verdes, 25 amarelos e 12 azuis. O número mínimo de lenços que devem ser retirados do cesto para que se possa garantir que, entre os lenços retirados, haja pelo menos quatro da mesma cor é:

- (a) 44.
- (b) 45.
- (c) 12.
- (d) 4.
- (e) 10.

**2.** Em programação é muito comum a criação de blocos de códigos, permitindo um melhor entendimento do código. Contudo essa técnica pode gerar certas problemáticas, como por exemplo, em relação ao escopo do bloco, ou seja, o que é permitido ser “visto” e utilizado pelo bloco, que implica onde deve ser declarado certas variáveis para que possam ser corretamente utilizadas. Analise o código a seguir:

---

Algoritmo

```

1.  Inicio
2.    Posicao1
3.    X ← Y + Z
4.    se (verdade) inicio
5.      Posicao2
6.      X ← Y - K + M
7.    fim_se
8.    X ← K*Y
9.  Fim

```

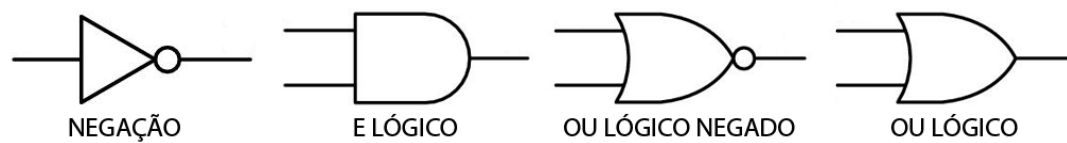
---

Algoritmo

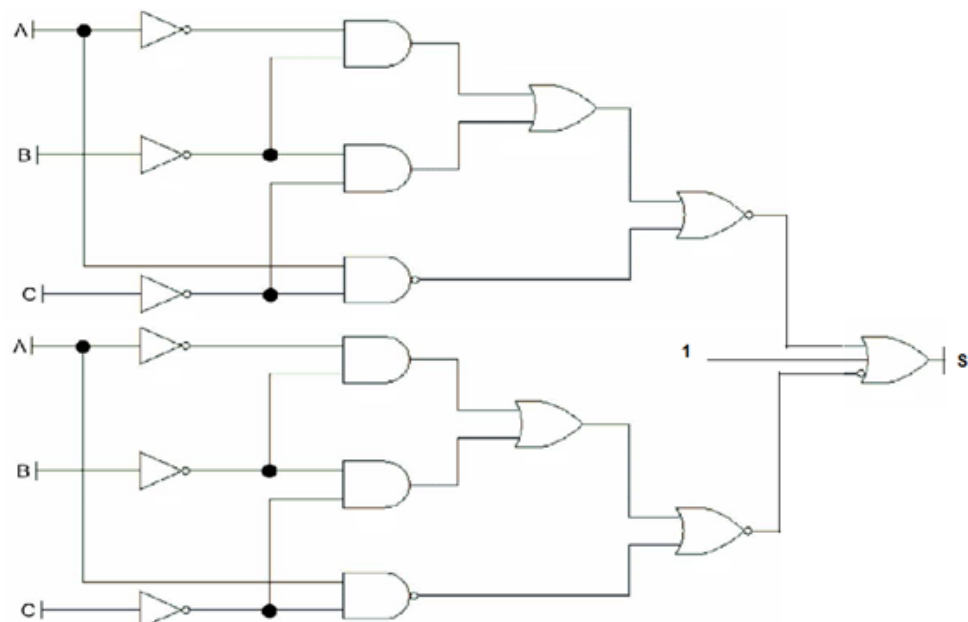
Sabendo que o escopo dos blocos é limitado ao que está dentro deles, ou seja, blocos fora dele não podem utilizar suas variáveis, para que o código funcione adequadamente, em qual opção as posições das variáveis X, Y, Z, K, M estão corretamente declaradas?

- (a) X – Posição2, Y – Posição1, Z – Posição2, K – Posição1, M – Posição1.
- (b) X – Posição1, Y – Posição1, Z – Posição1, K – Posição1, M – Posição2.
- (c) X – Posição1, Y – Posição1, Z – Posição1, K – Posição2, M – Posição1.
- (d) X – Posição1, Y – Posição1, Z – Posição1, K – Posição2, M – Posição2.
- (e) X – Posição1, Y – Posição2, Z – Posição1, K – Posição1, M – Posição2.

3. Sabendo que:



Considere a lógica booleana e o circuito lógico abaixo:



Assinale a alternativa que apresenta, corretamente, a expressão booleana minimizada para a saída S.

- (a)  $S = A\bar{B}\bar{C}$ .
- (b)  $S = 0$ .
- (c)  $S = 1$ .
- (d)  $S = \bar{A}\bar{B}\bar{C}$ .
- (e)  $S = ABC$ .

**4.** Dois colegas do curso de Computação, Fulano e Cicrano, estavam estudando para a prova de Algoritmos na semana seguinte. Resolvendo exercícios, os dois amigos depararam-se com um pseudocódigo muito peculiar:

---

Algoritmo

```

1.  Início
2.    val ← 0
3.    it ← 0
4.    b ← (-5)
5.    enquanto (val < 10) ou (b != 0) faca
6.      val ← val + 1
7.      it ← it + 1
8.      b ← b + 1
9.      escreve(b)
10.     escreve(val)
11.     escreve(it)
12.  fim_enquanto
13. Fim

```

---

Algoritmo

Pelo problema ser demasiadamente complicado, eles traduziram o programa para outra linguagem e executaram o programa no computador. O resultado de saída foi:

- (a) 5, 10, 10.
- (b) 10, 5, 5.
- (c) 0, 0, 5.
- (d) 10, 0, 5.
- (e) Não houve saída.

**5.** Considere a fórmula bem formada:

$$\forall x \forall y (Maior\_igual(media(x, y), x) \vee Maior\_igual(media(x, y), y))$$

De acordo com a lógica de predicados, a tradução da fórmula bem formada é:

- (a) O número x é maior ou igual do que a média de x e y considerando valores quaisquer.
- (b) Para todo x, há um y que é maior ou igual a média dos números considerados.
- (c) Existe um x que é maior ou igual a média de x e y dentro de um conjunto fechado a operação.
- (d) A média de quaisquer dois números é maior ou igual do que um dos dois
- (e) O número y é maior ou igual do que a média de x e y, considerando valores quaisquer.

**6.** Em programação estruturada, um dos conceitos aprendidos do método “dividir para conquistar” é a recursividade. A recursividade é a definição de uma função ou método que pode invocar a si mesma para desempenhar sua tarefa. Dessa forma, o uso de funções recursivas é altamente fundamental na estratégia de programadores para simplificar o código. Tendo isso em mente, responda qual dessas operações NÃO pode ser feita através do uso de função recursiva:

- (a) Cálculo do menor número de um vetor.
- (b) Cálculo do fatorial de um número.
- (c) Cálculo da potência de um número.
- (d) Exibição de conteúdo invertido de um vetor.
- (e) Cálculo de conversão de temperatura.

**7.** De acordo com o algoritmo abaixo em pseudocódigo:

---

Algoritmo

```

1.  Início
2.  M[10]: vetor de inteiro
3.  i: inteiro
4.  j: inteiro
5.  t: inteiro
6.  x: inteiro
7.  M ← { -14, 11, 32, 20, 15, 9, -8, 7, 33, 3 }
8.  para i ← 1 ate 9 faca início
9.      t ← i
10.     para j ← i+1 ate 10 faca
11.         se (M[j] < m[t]) entao
12.             t ← j
13.         x ← M[t]
14.     fim-para
15.     M[t] ← M[i]
16.     M[i] ← x
17.     para i ← 1 ate 10 faca
18.         escreva M[i]
19. Fim
```

---

Algoritmo

Assinale a alternativa que indica a saída padrão:

- (a) {3, 7, 9, 11, 15, 33, -8, -14, 20, 32}
- (b) {-14, -8, 3, 7, 9, 11, 15, 20, 32, 33}
- (c) {33, 32, 20, 15, 11, 9, 7, 3, -8, -14}
- (d) {3, -8, 7, -14, 9, 20, 11, 32, 15, 33}
- (e) {3, 7, -8, 9, 11, -14, 15, 20, 32, 33}

**8.** Considere a seguinte fórmula em lógica dos predicados:  $\forall x(A(x) \wedge \forall y(\neg B(x,y)) \rightarrow C(x))$ . Assumindo que o predicado  $A(x)$  significa que  $x$  é uma pessoa,  $B(x,y)$  significa que  $x$  é namorado de  $y$  e  $C(x)$  significa que  $x$  não é feliz, qual das alternativas abaixo melhor traduz a fórmula acima para o português?

- (a) Todas as pessoas que namoram com alguém são felizes.
- (b) Todas as pessoas que namoram com alguém não são felizes.
- (c) Todas as pessoas que não namoram com alguém são felizes.
- (d) Todas as pessoas que não namoram com alguém não são felizes.
- (e) Nenhuma das anteriores.

**9.** Um professor de lógica lançou aos seus alunos três sequências simples de afirmações:

1. Se você está olhando para muita areia, então você está vendo uma duna.  
Você está vendo uma duna.  
Logo, você está vendo muita areia.
2. Ricardo saltita quando está feliz.  
Ricardo está triste.  
Portanto, Ricardo não está saltitando.
3. Todo animal com polegares opositores opera instrumentos.  
Esse animal não tem polegares opositores.  
Assim, ele não pode operar instrumentos.

Ele, então, pediu para que seus alunos atestassem se elas eram verdadeiras ou não:

- (a) São todas verdadeiras.
- (b) 1 é falsa, 2 e 3 são verdadeiras.
- (c) 3 é a única verdadeira.
- (d) Apenas 2 é falsa.
- (e) São todas falsas.

**10.** Dada a sequência S abaixo e considerando uma leitura da esquerda para a direita da mesma, sempre que uma letra for lida ela deve ser inserida em uma pilha P e em uma fila F inicialmente vazias. Sempre que um caractere '\*' for lido, remova e imprima separadamente um caractere da pilha e da fila. Marque a opção que representa respectivamente as saídas para a pilha P e fila F ao final do processo.

S = "OL\*\*IMP\*IA\*\*\*\*DA\*CE\*\*AR\*\*\*EN\*SE\*\*\*"

- (a) Pilha: OLIMPIADACEARENSE e Fila: LOPAIMIAECRADNESE
- (b) Pilha: ESNEDARCEIMIAPOL e Fila: ESNERAECADAIPMILO
- (c) Pilha: OLIMPIADACEARENSE e Fila: ESNERAECADAIPMILO
- (d) Pilha: ESNERAECADAIPMILO e Fila: ESNEDARCEIMIAPOL
- (e) Pilha: LOPAIMIAECRADNESE e Fila: OLIMPIADACEARENSE

**11.** Analise o seguinte pseudo-código. Como está preenchido o vetor "saída" após uma execução do algoritmo? Considere que a primeira posição do vetor é 0.

---

Algoritmo

```

1.  Inicio
2.    i: inteiro
3.    j: inteiro
4.    k: inteiro
5.    entrada: vetor de inteiro
6.    u: vetor de inteiro
7.    saida: vetor de inteiro
8.    j <- 0
9.    k <- 0
10.   para i <- 0 ate 6 faca
11.     entrada[i] <- i + 1
12.   fim_para
13.   para i <- 0 ate 6 faca
14.     se (entrada[i] mod 2 = 0) entao
15.       p[j] <- entrada[i]
16.       j <- j + 1
17.     fim_se
18.     senao inicio
19.       u[k] <- entrada[i]
20.       k <- k + 1
21.     fim_senao
22.   fim_para
23.   para i <- 0 ate 3 faca
24.     saida[2*i] <- p[i]
25.     saida[(2*i) + 1] <- u[2 - i]
26.   fim_para
27. Fim

```

---

Algoritmo Ordem

- (a) {1, 2, 3, 4, 5, 6}
- (b) {6, 5, 4, 3, 2, 1}
- (c) {2, 5, 4, 3, 6, 1}
- (d) {1, 6, 3, 4, 5, 2}
- (e) {2, 4, 6, 5, 3, 1}

**12.** Um loop infinito é uma sequência de instruções de um programa de computador que se repete infinitamente, porque não há condição de parada ou porque a condição existe, mas nunca é atingida. Entre as opções abaixo, qual fragmento de algoritmo **não** é um loop infinito?

(a)

---

```
x <- 10
enquanto (x ≤ 10) faca
  x <- x - 1
fim_enquanto
```

---

Algoritmo A

(b)

---

```
x <- 2
enquanto (x > 0) faca
  escreva(x)
```

---

Algoritmo B

(c)

---

```
x <- 10
y <- 5
enquanto (x + y) != (x - y) faca
  x <- x - y
  y <- y - 1
fim_enquanto
```

---

Algoritmo C

(d)

---

```
x <- 1
y <- 10
enquanto (y != x) faca
  x <- x + 2
  y <- y - 2
fim_enquanto
```

---

Algoritmo D

(e) Todas as opções acima são loops infinitos.

**13.** Existem 4 bolas: A, B, C e D. Cada uma irá ser pintada por Daniel e Rairis por cores distintas: preto, laranja, verde e amarelo. Com isso, Marcos terá que adivinhar a cor de pelo menos uma das bolas, baseado nas afirmações do casal, sendo que Daniel sempre mente e Rairis sempre fala a verdade. Seguem as informações:

- Rairis: A é verde se e somente se C for amarelo.
- Daniel: Se C for laranja então D é amarelo.
- Rairis: Se D for preto então A não pode ser amarelo.

Assim, Marcos pode afirmar que:

- (a) D é verde.
- (b) A é preto.
- (c) B é amarelo.
- (d) A é amarelo.
- (e) D é preto.

**14.** Expressões lógicas são aquelas cujo resultado da avaliação é um valor lógico (verdadeiro ou falso). Considere as expressões abaixo.

1.  $(6 < 8) \vee (3 > 7)$
2.  $\neg(6/2 >= 3) \wedge (7 >= (12/3 + (3 * 2))) = \neg(5 > 3 \wedge 2 >= 1)$
3.  $(5 >= 6 \wedge 6 < 7) \vee \neg(a + 5 - 6 = 8) \text{ \{onde } a = 5\}$
4.  $((34 > 9 \wedge 5 + u = 34) \vee (5 = 15/3 \wedge 8 > 12)) = ((u = 29) \wedge 8 > 12) \text{ \{onde } u = 29\}$

Os resultados verdadeiros correspondem às avaliações das expressões lógicas em:

- (a) 3.
- (b) 2.
- (c) 2 e 3.
- (d) 1 e 4.
- (e) 1 e 3.

**15.** Recursividade é um conceito amplamente usado na computação. Uma função é dita recursiva quando invoca a si mesma durante sua execução. Sabendo que a função a seguir é recursiva, qual é o valor de  $F(7)$ ?

$$\begin{aligned} F(0) &= 1 \\ F(1) &= 1 \\ F(x) &= F(x-1) \times F(x-2), \text{ para } x > 2 \end{aligned}$$

- (a) 8.
- (b) 5.
- (c) 13.
- (d) 7.
- (e) 10.

**16.** Dois amigos querem descobrir quantos degraus estão expostos em uma escada-rolante que sobe. Para isso, os dois concordam em subir e contar o número de passos dados. O primeiro, achando ótima a ideia, começou a subir a escada rolante, degrau por degrau. O segundo amigo percebeu imediatamente que o plano não iria funcionar se eles subissem ambos desse modo. Partindo junto com o primeiro, pôs-se a subir de dois em dois degraus, com o dobro da velocidade do amigo. No final, o primeiro disse ter dado 21 passos, enquanto o segundo afirmou ter dado 15. Após alguns pensamentos de lógica e poucas contas matemáticas, concordaram que a escada tinha, totalmente expostos:

- (a) 30 degraus.
- (b) 52 degraus.
- (c) 48 degraus.
- (d) 36 degraus.
- (e) 42 degraus.

**17.** Numere de 1 à 8 os termos a seguir, de acordo com a sequência de passos que levam à resolução de um problema usando computador.

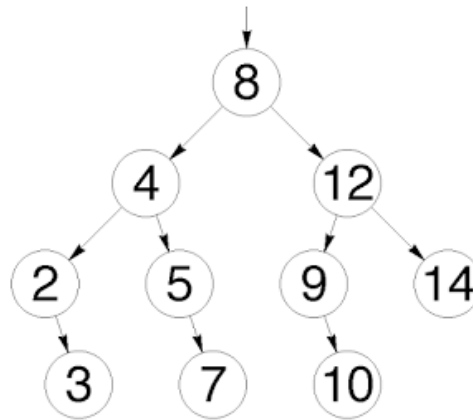
- ( ) programa em linguagem de máquina
- ( ) definição do problema
- ( ) definição da linguagem de alto nível
- ( ) compilação e montagem
- ( ) elaboração do algoritmo
- ( ) programa em linguagem de alto nível
- ( ) execução
- ( ) carregamento na memória

- (a) 3-1-2-7-4-5-6-8.
- (b) 5-4-2-6-1-3-7-8.
- (c) 3-2-4-5-7-6-8-1.
- (d) 6-1-3-5-2-4-8-7.
- (e) 7-1-5-3-2-6-4-8.

**Texto para as questões 18, 19 e 20.**

**Lorma-Caia** é uma estrutura de dados composta por uma árvore binária de busca de arranjos. **Árvore binária** de busca é uma estrutura de dados baseada em nós, onde todos os nós da subárvore esquerda possuem um valor numérico inferior ao nó raiz e todos os nós da subárvore direita possuem um valor superior ao nó raiz.

Exemplo:



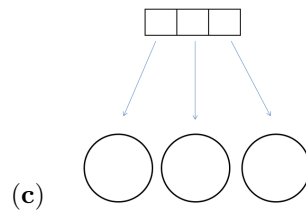
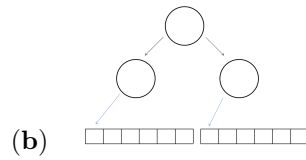
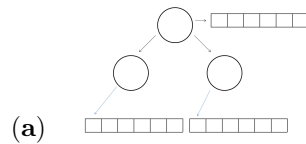
**Array, arranjo** ou **variável indexada**, também conhecido como vetor (para arrays uni-dimensionais) ou matriz (para arrays bi-dimensionais), é uma das mais simples estruturas de dados, onde elementos individuais são acessados por sua posição no arranjo, a qual é dada por um índice.

Exemplo de arranjo: veja que a posição 0 tem o valor 5, a posição 1 tem o valor 10 e assim segue o raciocínio:

Valor da Celulas						
Indices	5	10	20	1	0	5
	0	1	2	3	4	5

Na nossa estrutura híbrida um elemento da árvore contém um arranjo, onde todos os seus elementos são menores ou iguais do que o valor indicado em seu nó. Logo, um arranjo no nó de valor 3 não pode ter elementos maiores que 3.

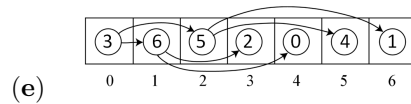
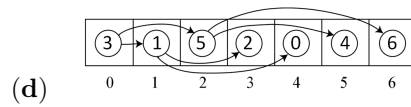
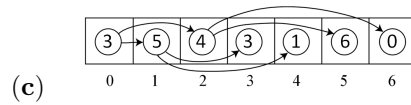
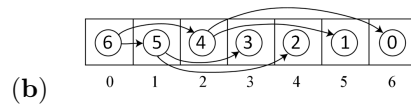
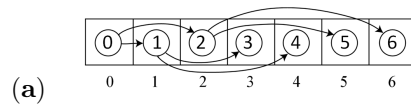
**18.** Qual das figuras a seguir melhor representaria a definição de uma Lorma-Caia?



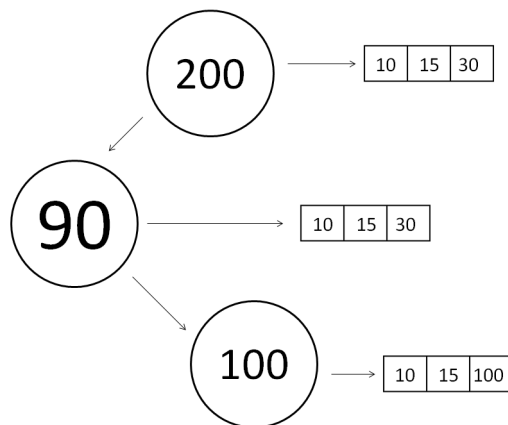
(d) As três formas anteriores são corretas.

(e) Nenhuma das formas é correta.

**19.** Qual das figuras a seguir melhor representaria uma árvore binária de busca, implementada como um arranjo?



**20.** O comando  $\text{Move}[\text{Nó}, \text{Índice}]$  retorna o valor presente no índice do arranjo presente no nó da **Lorma-Caia**, ou seja,  $\text{Move}[10, 10]$  deve retornar o valor presente no índice 10 do vetor contido no nó com valor 10. O comando  $\text{Move}[\text{Nó}, \text{null}]$  retorna o valor presente no nó. Assinale o que se pode afirmar sobre a seguinte figura abaixo, para tal, considere os índices dos arranjos partindo de zero.



- (a) Ela não representa uma Lorma-Caia.
- (b) O comando  $\text{Move}[100, \text{null}]$  retorna um valor equivalente à  $\text{Move}[100, 2]$ .
- (c) O comando  $\text{Move}[100, 1]$  retorna 100.
- (d) O comando  $\text{Move}[\text{null}, \text{null}]$  retorna 100.
- (e) O comando  $\text{Move}[200, 10]$  retorna 10.

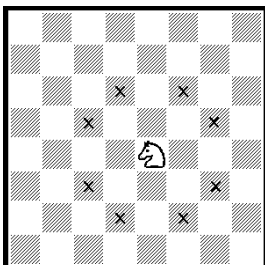
**21.** Define-se **mediana** como o número  $x$  tal que  $\exists y \in \{a, b, c\}$ , onde  $y \leq x$  e  $\exists z \in \{a, b, c\}$ , onde  $z \geq x$ . Dadas as funções:

1.  $\text{Min}(x, y)$ : retorna  $x$ , se  $x < y$  e  $y$  caso o contrário.
2.  $\text{Max}(x, y)$ : retorna  $x$ , se  $x > y$  e  $y$  caso o contrário.
3.  $\text{Média}(a, b, c)$ : retorna  $(a + b + c)/3$
4.  $\text{Moda}(a, b, c)$ : retorna o número mais presente nos termos.

Com quantas e quais funções são necessárias para se definir a mediana.

- (a) 2, Min e Max.
- (b) 3, Min, Max e Moda.
- (c) 3, Min, Max e Média.
- (d) 4, todas as funções.
- (e) 2, Média e Moda.

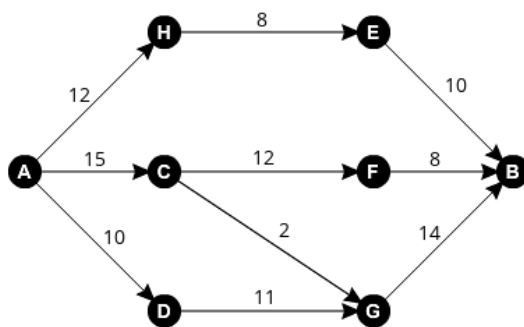
**22.** O Xadrez como é jogado atualmente é um jogo de tabuleiro ocidental que surgiu na segunda metade do século XV, durante o Renascimento, depois de ter evoluído de suas antigas origens persas e indianas. Nele, dois jogadores se enfrentam em um tabuleiro quadrado dividido em 64 casas ( $8 \times 8$ ). Cada jogador possui 6 tipos de peças que se movimentam de maneiras próprias. Uma destas peças é o cavalo, que se movimenta deslocando-se em “L” no tabuleiro, ou seja, movendo-se duas casas para uma direção e uma casa numa direção perpendicular.



Assim sendo, qual a quantidade máxima de cavalos que é possível distribuir em um tabuleiro  $7 \times 7$  sem que nenhum cavalo fique ameaçando outro, ou seja, sem que nenhum cavalo fique em uma posição a qual é destino do movimento de outro cavalo?

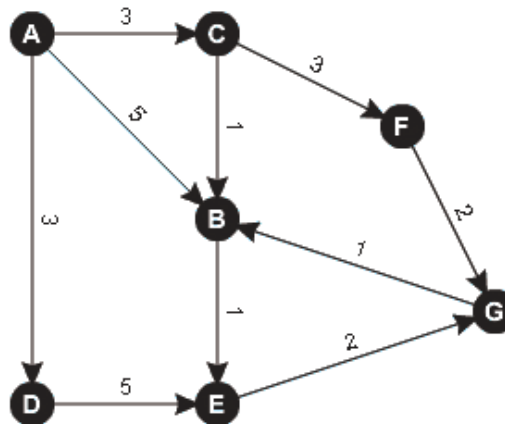
- (a) 15.
- (b) 18.
- (c) 21.
- (d) 25.
- (e) 28.

**23.** Considerando a imagem abaixo como um sistema de encanamento que leva água do ponto A ao B e o peso de cada cano sendo o volume em litros que cada uma pode suportar, qual o máximo de água, em litros, que pode chegar ao ponto B de uma só vez?



- (a) 32.
- (b) 31.
- (c) 30.
- (d) 29.
- (e) 28.

**24.** Um grafo é um conjunto de vértices e arestas, no qual uma aresta representa uma relação entre dois vértices na forma de pares ordenados e cada relação pode ter um valor (chamado de peso). Em um grafo direcionado, como o da figura, essa relação é unilateral (vai do vértice de partida ao vértice de chegada, por exemplo,  $A \rightarrow B$  representa a relação  $(a,b)$ , mas a relação  $(b,a)$  não vale). Um caminho é um conjunto de vértices ordenados de maneira que um vértice sempre tem uma aresta para o próximo vértice. Para saber o peso de um caminho entre dois vértices, soma-se o peso das arestas percorridas do vértice inicial até o final. O caminho entre dois vértices que tem o menor peso é chamado de menor caminho.



Qual o menor caminho entre A e G?

- (a) 4.
- (b) 5.
- (c) 6.
- (d) 7.
- (e) 8.

**25.** Existem dois tipos de heaps: Os heaps de máximo (max-heap), em que o valor de todos os nós são menores que os de seus respectivos pais; e os heaps de mínimo (min-heap), em que o valor de todos os nós são maiores que os de seus respectivos pais. Assim, em um heap de máximo, o maior valor do conjunto está na raiz da árvore, enquanto no heap de mínimo a raiz armazena o menor valor existente. Após um processo de ordenação, indique qual deve ser saída para que o vetor abaixo configure corretamente a estrutura de dados denominada max-heap.

Vetor: = {10, 2, 17, 3, 19, 6, 23, 7, 9, 35, 11}

- (a) {35, 23, 19, 17, 11, 2, 3, 6, 7, 9, 10}.
- (b) {2, 3, 6, 7, 9, 10, 35, 23, 19, 17, 11}.
- (c) {2, 3, 6, 7, 9, 10, 11, 17, 19, 23, 35}.
- (d) {10, 9, 7, 6, 3, 2, 35, 23, 19, 17, 11}.
- (e) {35, 19, 23, 17, 11, 9, 10, 7, 6, 2, 3}.