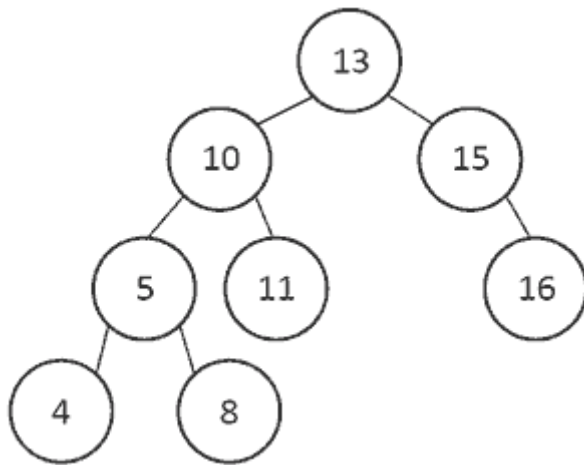


Uma árvore AVL é um tipo de árvore binária balanceada na qual a diferença entre as alturas de suas subárvores da esquerda e da direita não pode ser maior do que 1 para qualquer nó. Após a inserção de um nó em uma AVL, a raiz da subárvore de nível mais baixo no qual o novo nó foi inserido é marcada. Se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

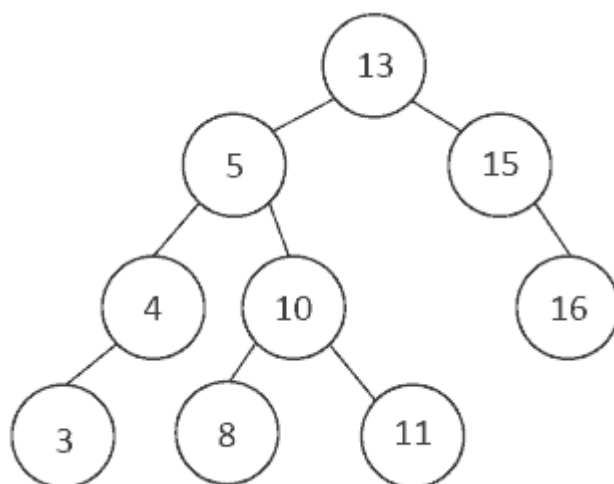
LAFORE, R. **Data Structures & algorithms in Java**. Indianópolis: Sams Publishing, 2003 (adaptado).

A seguir, é apresentado um exemplo de árvore AVL.



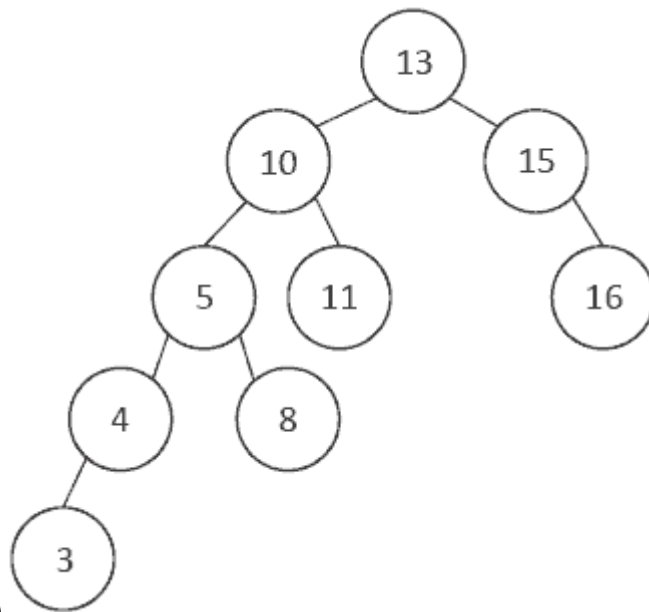
Árvore AVL da questão 09

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



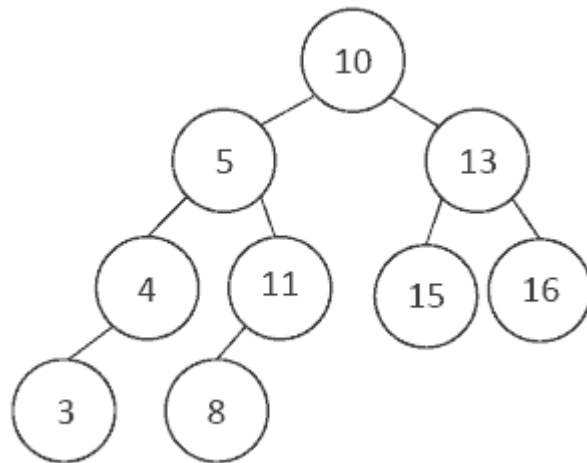
- (A)

Alternativa A



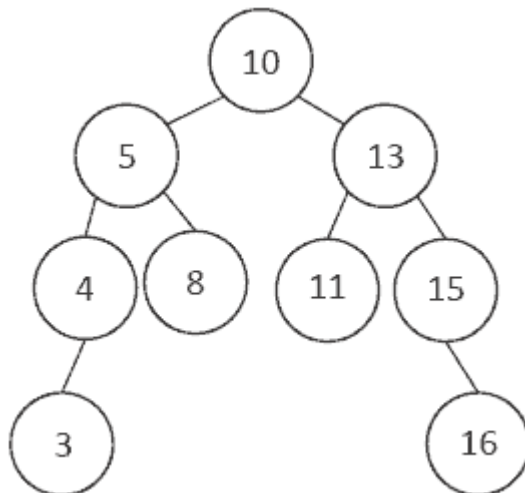
• (B)

Alternativa B



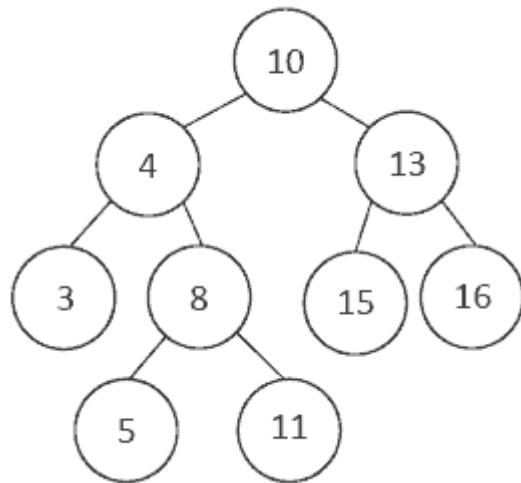
• (C)

Alternativa C



• (D)

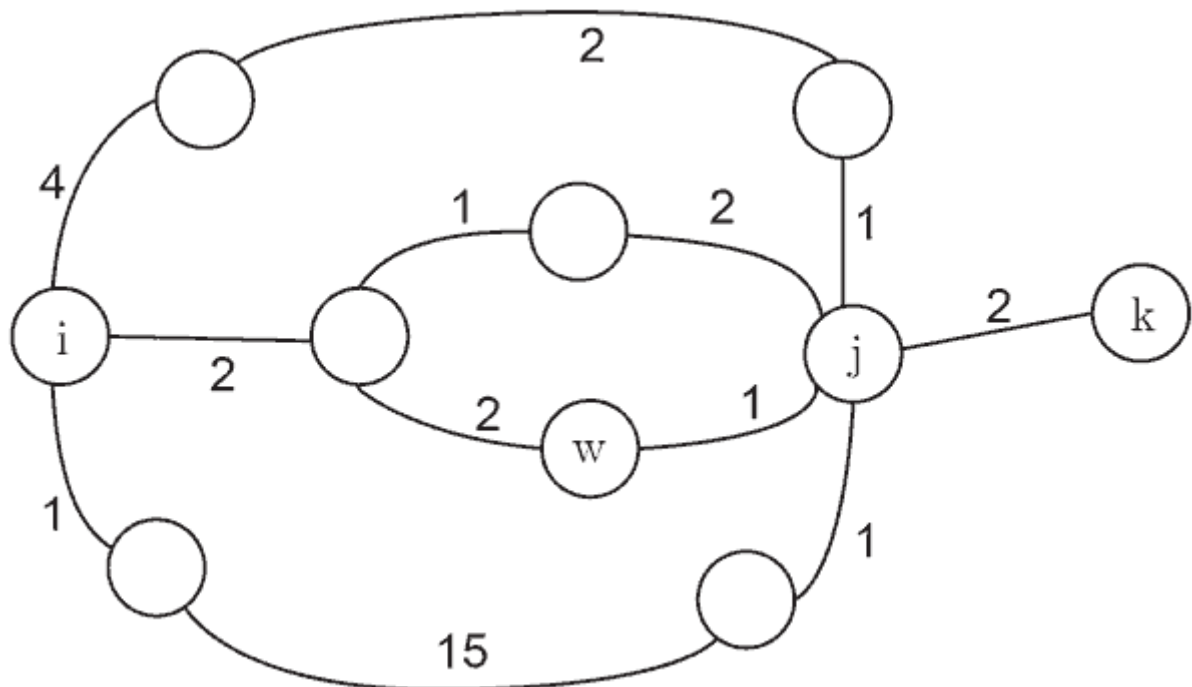
Alternativa D



• (E)

Alternativa E

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.

- 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.

A pilha é uma estrutura de dados que permite a inserção e a remoção desses dados sempre por meio de regras predefinidas. Para que essas operações sejam realizadas, são utilizadas duas funções: push e pop. Com base nessa informação, considere que um programa possua uma pilha p , inicialmente vazia, e que as seguintes operações foram realizadas: PUSH(p , 10); PUSH(p , 5); PUSH(p , 3); PUSH(p , 50); POP(p); PUSH(p , 11); PUSH(p , 9); PUSH(p , 20); POP(p); POP(p).

Ao fim da execução desses comandos, quais serão o topo da pilha e o somatório dos elementos ainda dentro da pilha, respectivamente?

A50 e 68.

B20 e 58.

C11 e 29.

D9 e 38.

E3 e 29.

```

algoritmo Pilha
...
inicio
    IniciarPilha(s)
    enquanto (não for o final das entradas) faça
        leia num
        se (num <> 3)
            entao Empilhar(s, num)
            senao Desempilhar(s)
             $x \leftarrow \text{ElementoTopo}(s)$ 
        fimse
    fimenquanto
fimalgoritmo

```

Considere que, no trecho do programa acima, representado por seu pseudocódigo, seja fornecido para num, sucessivamente, os valores inteiros 1, 2, 3, 4, 5, 3 e 6. Nesse caso, ao final da execução do programa, o valor de x será igual a

- A2 e a pilha terá os valores 6, 4 e 1.
- B3 e a pilha terá os valores 6, 4 e 1.
- C5 e a pilha terá os valores 6, 4 e 1.
- D3 e a pilha terá os valores 6, 5, 4, 2 e 1.
- E5 e a pilha terá os valores 6, 3, 5, 4, 3, 2 e 1.