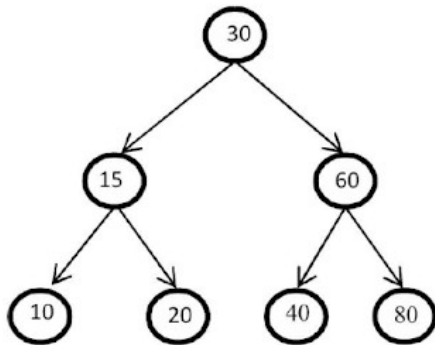


EXERCÍCIO - Aula 05 – Árvores Reconstrução e BST

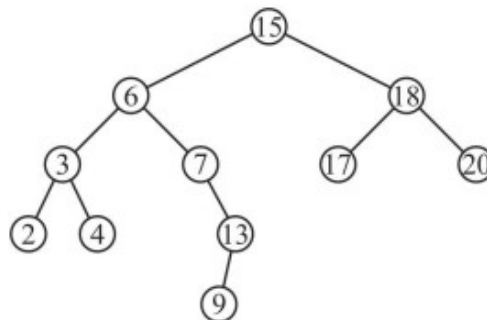
1) Observe a Árvore de Busca Binária (ABB) a seguir.



Assinale a alternativa que apresenta, corretamente, a sequência de inserção que gera essa ABB.

- a) 30, 15, 40, 10, 20, 60, 80
- b) 30, 15, 40, 10, 20, 80, 60
- c) 30, 15, 60, 10, 20, 40, 80
- d) 30, 60, 20, 80, 15, 10, 40
- e) 30, 60, 40, 10, 20, 15, 80

2) Analise a árvore a seguir e responda as questões 2.1 e 2.2:



Thomas H. Cormen *et al.* Algoritmos: teoria e prática. Editora Campus, v. 2, 2002. p. 207.

2.1) Mostre os percursos

Pre-ordem:

In-Ordem:

Pós-ordem:

2.2) De acordo com a figura anterior, o procedimento

```

CONSULTA (x)
1  while esquerda [x] ≠ NIL
2    do x ← esquerda [x]
3  return x
  
```

realiza, na árvore, a consulta de

- a) search.
- b) minimum.
- c) maximum.
- d) successor.
- e) predecessor.

3) Reconstrua graficamente as árvores a partir dos percursos **pré-ordem** e **in-ordem** e em seguida apresente o percurso em **pós-ordem**.

3.1)

Pré: 1 6 7 8

In: 1 6 8 7

Pós:

3.2)

Pré: 0 1 3 4 2 5

In: 3 1 4 0 5 2

Pós:

3.3)

Pré: A B D E C F

In: D B E A F C

Pós:

3.4)

Pré: 40 25 20 10 15 23 32 28 82 55 90 87 100

In: 10 15 20 23 25 28 32 40 55 82 87 90 100

Pós:

4) Tendo como base o algoritmo de árvore binária de busca visto em sala, implemente os seguintes métodos:

- **altura()** : retorna a altura tendo como base o nó passado como referencia
- **minimo()** : retornar o menor valor da arvore
- **maximo()** : retornar o maior valor da arvore
- **remocao()** : implemente um método que remova um elemento passado como parâmetro