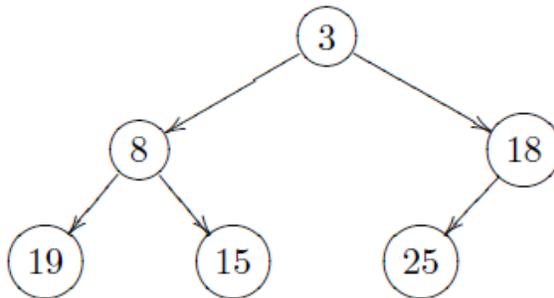


**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO NORTE
Campus Natal - Central

Exercícios de Estrutura de Dados – Heap

- Desenhe uma Árvore binária completa (balanceada) que contenhas as seguintes chaves: {5, 6, 10, 11, 30, 28, 15, 9, 14, 13, 7, 8, 20, 19, 12, 23, 25} e que as mesmas sejam apresentadas nesta ordem considerando uma travessia pré ordem e responda:
 - Esta árvore é um heap? Se não for, modifique-a para que seja.
 - Mostre os passos para a inserção de um item com a chave 16 neste heap e depois faça o desenho do novo heap
 - Mostre os passos para a remoção do item com chave 15 deste heap.
- Indique se a frase a seguir é verdadeira ou falsa, justificando: “Em um heap implementado usando arrays, o elemento com a maior chave está no maior índice usado do array.”
- Sobre a figura abaixo:



Desenhe o heap durante e após cada operação a seguir:

- Inserção da chave 11
 - removeMin
 - removeMin
 - removeMin
 - Inserção da chave 31
 - Inserção da chave 1
- Ilustre a execução do algoritmo heap-sort para a seguinte sequencia: {2, 5, 16, 4, 23, 39, 26, 15}. Desenhe o conteúdo tanto da Sequencia como da Fila de Prioridade para cada passo do algoritmo, incluindo os passos do restauração da heap-order no heap.
 - Sobre heap, considere a interface a seguir:

```
public interface heap extends ArvoreBinaria
{
    public void insert(Object chave, Object elemento);
    public Item remove();
}
```

- a) Escreva uma classe que implemente heap usando um nó que armazene o pai e os filhos da esquerda e da direita, além de uma referência para um elemento.
- b) Escreva uma classe que implemente heap usando um array.