

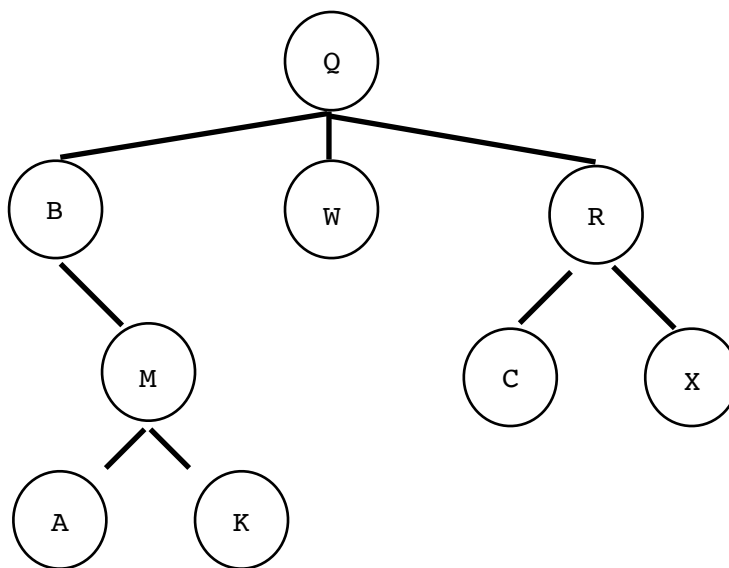
**Centro de Ensino Superior de Juiz de Fora**  
**Estruturas de Dados para Pesquisa e Ordenação**  
**Prof.a. Bárbara Quintela**

**Lista de Exercícios para Segunda Prova**

**Questao 1. (a)** Desenhe a árvore binária de busca se forem inseridos os seguintes elementos na ordem dada: 12 15 3 35 21 42 14

**(b)** Desenhe a árvore binária de busca *balanceada* contendo os mesmos números dados na letra (a)

**Questão 2.** Considere a seguinte árvore que **não** é binária:



(a) Qual é ou quais são o(s) nó(s) raiz?

(b) Qual é ou quais são o(s) nó(s) folha(s)?

(c) Escreva abaixo os nós na ordem que seriam percorridos caso seja feito um caminhamento em **pós ordem** dessa árvore começando no nó Q:

(d) Escreva abaixo os nós na ordem que seriam percorridos caso seja feito um caminhamento em **pré ordem** dessa árvores começando no no Q:

**Questão 3. Dado o algoritmo de busca sequencial em Python abaixo, Responda:**

```
def orderedSequentialSearch(alist, item):
    pos = 0
    found = False
    stop = False
    while pos < len(alist) and not found and not stop:
        if alist[pos] == item:
            found = True
        else:
            if alist[pos] > item:
                stop = True
            else:
                pos = pos+1
    return found

testlist = [0, 1, 2, 8, 13, 17, 19, 32, 42,]
print(orderedSequentialSearch(testlist, 3))
print(orderedSequentialSearch(testlist, 13))
```

(a) Suponha que você esteja fazendo uma busca sequencial da lista [15, 18, 2, 19, 18, 0, 8, 14, 19, 14]. Quantas comparações precisa fazer para encontrar a chave 18?

- I. 5
- II. 10
- III. 4
- IV. 2

(b) Suponha que você esteja fazendo uma busca sequencial da lista ordenada [3, 5, 6, 8, 11, 12, 14, 15, 17, 18]. Quantas comparações precisa fazer para encontrar a chave 13?

- I. 10
- II. 5
- III. 7
- IV. 6

**Questão 4. Dado o código Python de busca binária recursiva abaixo, responda:**

```
def binarySearch(alist, item):
    if len(alist) == 0:
        return False
    else:
        midpoint = len(alist)//2
        if alist[midpoint]==item:
            return True
        else:
            if item<alist[midpoint]:
                return binarySearch(alist[:midpoint],item)
            else:
                return binarySearch(alist[midpoint+1:],item)

testlist = [0, 1, 2, 8, 13, 17, 19, 32, 42,]
print(binarySearch(testlist, 3))
print(binarySearch(testlist, 13))
```

(a) Suponha que voce possui a seguinte lista ordenada [3, 5, 6, 8, 11, 12, 14, 15, 17, 18] e está usando o algoritmo de busca binária recursiva acima. Qual grupo de números mostra a sequencia de comparações utilizada para encontrar a chave 8:

- (I) 11, 5, 6, 8
- (II) 12, 6, 11, 8
- (III) 3, 5, 6, 8
- (IV) 18, 12, 6, 8

(b) Suponha que voce possui a seguinte lista ordenada [3, 5, 6, 8, 11, 12, 14, 15, 17, 18] e está usando o algoritmo de busca binária recursiva acima. Qual grupo de números mostra a sequencia de comparações utilizada para encontrar a chave 16:

- (I) 11, 14, 17
- (II) 18, 17, 15
- (III) 14, 17, 15
- (IV) 12, 17, 15