

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DO RN
 GERÊNCIA DE TECNOLOGIA DA INFORMAÇÃO E EDUCACIONAL DE TELEMÁTICA
 Exercícios de Estrutura de Dados I

1. Sobre a árvore da figura 1, responda:

- (a) Qual é o nó raiz?
- (b) Quais são os nós internos?
- (c) Quantos descendentes tem o nó g?
- (d) Quantos são os ancestrais de n?
- (e) Quais são os irmãos de k?
- (f) Que nós pertencem a subárvore com raiz no nó b
- (g) Qual a profundidade do nó p?
- (h) Qual a altura da árvore?
- (i) Qual o resultado de uma impressão pré ordem?
- (j) Qual o resultado de uma impressão pós ordem?

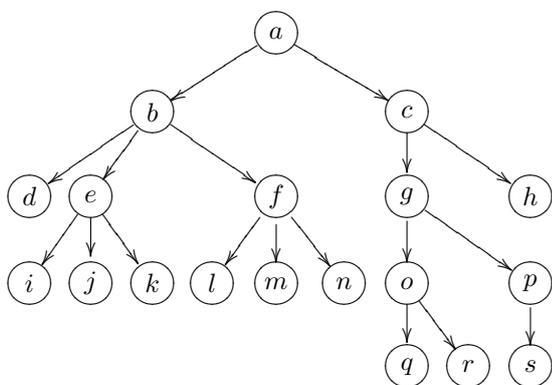


Figura 1: Árvore genérica

2. Ainda sobre árvores genéricas: baixe os arquivos `Position.java`, `InvalidPositionException`, `Arvore.java`, `ArvoreGenerica` e `ArvoreSimples.java`, entenda-os e:

- (a) Implemente os métodos:
 - `replace(Object o)`
 - `swapElement()` e
- (b) Refaça o método `depth()` de forma iterativa (atualmente ele é recursivo).
- (c) Implemente o método `height()`.
- (d) Implemente um método (`preorderPrint`) que retorne uma **String** contendo os nós (fazendo a travessia pré ordem), separados por vírgula. Use o método `toString()` de `Object`.
- (e) Implemente um método (`posorderPrint`) que retorne uma **String** contendo os nós (fazendo a travessia pós ordem) separados por vírgula. Use o método `toString()` de `Object`.

3. Desenvolva um algoritmo que conte o número de nós de uma árvore. Considere a existência de uma classe `NoArvore` com os seguintes atributos:

- `private Object o;`
- `private NoArvore pai;`
- `private Vector filhos;`

4. Descreva quais exceções podem ser disparadas pelos métodos de `ArvoreSimples` e implemente-as.

5. Podemos representar uma árvore usando um objeto `String`, isto é chamado de **Representação usando parênteses** (*Parenthetic Representation*). Esta representação usa o nó em forma de `String` seguido pela representação dos seus filhos entre parênteses, separados por vírgula. Por exemplo, a árvore da figura 1 seria:

$$a(b(d, e(i, j, k)f(l, m, n)), c(g(o(q, r), p(s)), h))$$

Seja T uma árvore, a representação usando parênteses $P(T)$ de T pode ser definida recursivamente como:

- (a) Se T possui apenas um nó v , então $P(T) = v.element().toString()$, ou
- (b) $P(T) = v.element().toString() + "(" + P(T_1) + "," + P(T_2) + "," + \dots + P(T_k) + ")"$ onde v é a raiz de T e $T_1 \dots T_k$ são as subárvores de T com raiz nos filhos de T .

Partindo dessa definição, implemente um método `public String parentheticRepresentation()` que retorne uma `String` com a representação usando parênteses da árvore.

6. Desenvolva um algoritmo para criar uma nova árvore a partir de uma `String` que seja uma representação usando parênteses de uma árvore.

- Dica 1: Pode ser um construtor da árvore
- Dica 2: É necessário usar uma pilha.

7. Sobre árvore binárias:

- (a) Baixe a interface `ArvoreBinaria`, que estende de `Arvore`
- (b) Implemente usando arranjos
- (c) Implemente usando nós ligados

8. Desenhe uma representação sob a forma de uma árvore binária da árvore da figura 1.

9. Faça um método em `ArvoreSimples` que retorne uma Árvore binária contendo os elementos da árvore em representação binária.

10. Desenvolva um fórum discussão onde as mensagens fiquem armazenadas numa árvore, onde os nós são as mensagens. A raiz é uma mensagem padrão do fórum e as respostas armazenadas como nó filho da mensagem respondida. Crie métodos para armazenar a árvore no disco e para ler a árvore do disco. Faça o fórum usando servlets.