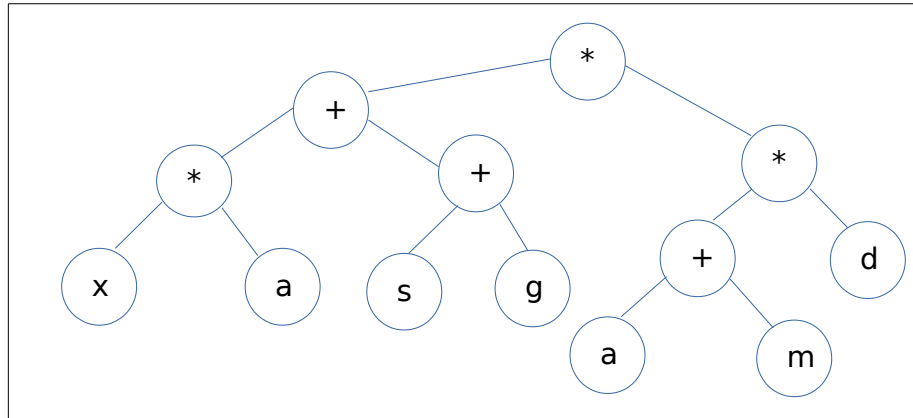


#### 4. Evaluar expresión aritmética (2,5 puntos)

Queremos hacer un subprograma que tome como entrada un árbol que representa una expresión aritmética y una tabla hash que contiene los valores de las variables en un momento dado de la ejecución del programa.

La expresión aritmética puede contener en sus nodos dos tipos de elementos: nombres de variable u operadores (solo se admitirán los operadores de suma y multiplicación).



El árbol anterior representa la expresión:  $((x * a) + (s + g)) * ((a + m) * d)$

Además, se tiene una tabla hash que contiene las variables con sus valores:

x	4
a	5
s	7
g	1
m	5
d	2

El subprograma deberá devolver el valor numérico resultado de evaluar esa expresión.

En el ejemplo dado, el resultado es  $560 = ((4 * 5) + (7 + 1)) * ((5 + 5) * 2)$

```
public class BinaryTreeNode<T> {
    T element;
    BinaryTreeNode<T> left, right;
}

public class ArbolExpresion {
    BinaryTreeNode<InfoElemExp> root;

    public Integer evaluar(HashMap<String, Integer> tHash)
    // pre: tHash contiene los valores de las variables
    // post: Se ha evaluado la expresión correspondiente al árbol
    //       Los valores de las variables se han tomado de tHash
    //       Si una variable del árbol no se encuentra en la
    //       tabla hash, se asumirá que el valor por defecto es cero
}

public class InfoElemExp {
    String elem; // *, +, ó nombre de una variable
    boolean operador; // true -> operador, false -> variable
}
```

Se pide:

- Implementar el algoritmo
- Establecer de manera razonada el coste del algoritmo.