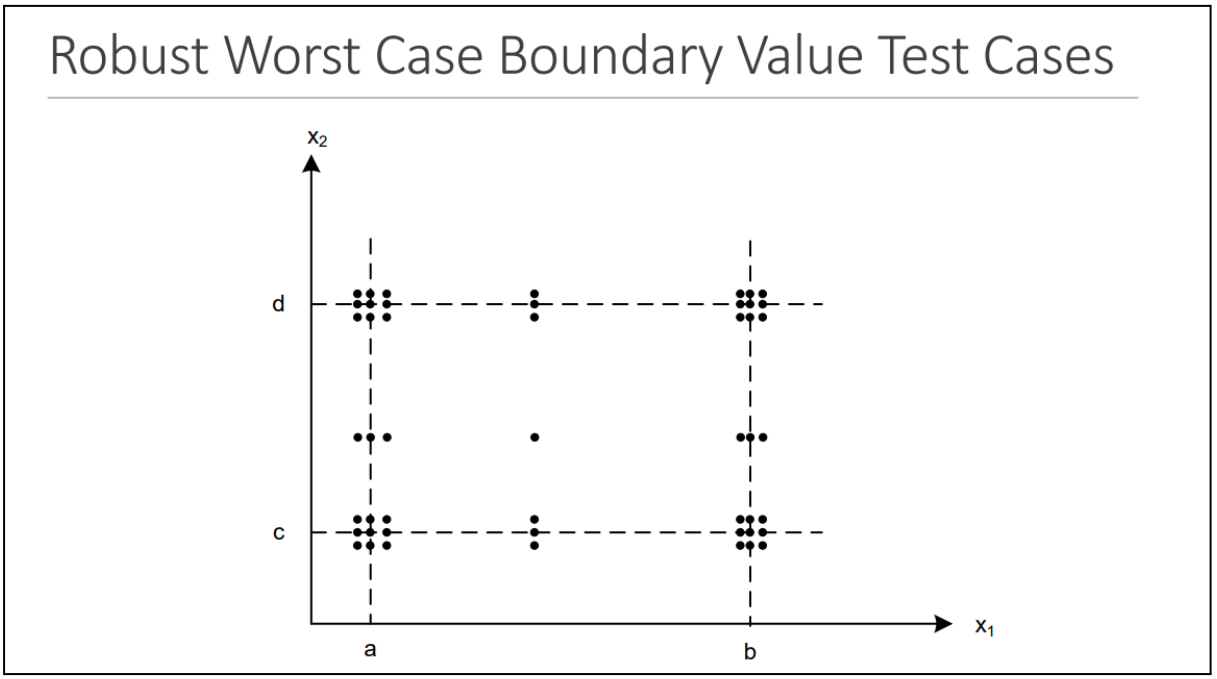
2. Las siguientes funciones se probarán empleando pruebas de valor límite:
a. insert
b. search
c. printBetweenLevel

El árbol binario descrito recibe como entrada números enteros. Los valores de los nodos deben encontrarse en el rango [-2500,2500). Además, no se permitirán árboles con una profundidad mayor a 50 elementos.

Estas pruebas deben evaluar las dos condiciones previamente mencionadas. Para ello, se usará una aproximación Worst Case Robust Boundary Value Testing.



Robust "Worst Case" Boundary Value Testing: 7^n

Siendo n el número de variables independientes

a. insert

insert

```
public void insert(T content,
                   boolean recursive)
            throws com.exceptions.DepthException
```

Description copied from class: `AbstractBST`

Inserta un elemento en el árbol

Specified by:

`insert` in class `AbstractBST<T extends Comparable<T>>`

Parameters:

`content` - New Content to insert in the tree.

`recursive` - true if method is recursive, false otherwise

Throws:

`com.exceptions.DepthException` - si intentamos insertar un elemento a una profundidad mayor que la permitida

Test
Numero de variables= 2, Rango[-2500,2500), Profundidad [1,50], en recursive pongo
7^2=49, 49*2=98
b. search

search

```
public Node<T> search(T content)
```

Description copied from class: `AbstractBST`

Method to find a piece of content. If the method not find the piece return null

Specified by:

`search` in class `AbstractBST<T extends Comparable<T>>`

Parameters:

`content` - Generic content to find, must implement the .equals() function.

c. printBetweenLevel

printBetweenLevel

```
public List<T> printBetweenLevel(int a,
                                 int b)
                          throws com.exceptions.BetweenLevelException
```

Specified by:

`printBetweenLevel` in class `AbstractBST<T extends Comparable<T>>`

Parameters:

a - nivel desde

b - nivel hasta

Returns:

una lista con todos los elementos que hay entre los niveles a y b, ambos incluidos

Throws:

`com.exceptions.BetweenLevelException` - El nivel a debe ser mayor que b, en caso contraio lanzará una BetweenLevelException. Ninguno de los dos niveles debe ser mayor que la profundidad máxima del árbol, en caso contrario lanzará una BetweenLevelException. Ninguno de los dos niveles puede ser cero o menor que cero, en caso contrario lanzará BetweenLevelException