Algorithm

isExclusive(A, B, C)

C contain all element. in A or B not A and B

Dx := new Dict(HT)

Load.Dictionary(A, A.first(), Dx)

Compute XOR.(B, p, Dx)

if !c.isEmpty() then.

DC := new Dict(HT)

Load Dictionary(C, C.first(), C, DC)

~~if~~ Dx.size() ≠ Dc.size() the..

return false.

else. iterX := Dx..keys().

CheckXOR(iterX, DC)

(1, a) A

(3, b) B

2, A = 2.

1, (a)

3, (b)

A

C = {(1)(1), 2}

B = {1, 2}

< 1, ~~2~~ C 1 |

0 . 0

return

Algorithm computeXOR(B, p, Dx)

be := p.element()

v := Dx.findValue(be)

if v = a. then

Dx.remove(b)

Algorithm CheckXOR(iterX, DC).

```
e := iterX.next().
v := DC.findValue(e).
if v = NO_SUCH_KEY then.
    return false.
if iter.hasNext() then.
    return checkXOR(iter, DC).
else.
    return true.
```