```
input  : A bitmap $Im$ of size $w \times l$
output: A partition of the bitmap
special treatment of the first line
for $i \leftarrow 2$ to $l$ do
    special treatment of the first element of line $i$
    for $j \leftarrow 2$ to $w$ do
        left ← FindCompress($Im[i, j-1]$)
        up ← FindCompress($Im[i-1, ]$)
        this ← FindCompress($Im[i, j]$)
        if left compatible with this then /* O(left,this)==1          */
            if left < this then Union(left,this)
            else Union(this,left)
        end
        if up compatible with this then                    // O(up,this)==1
            if up < this then Union(up,this)
            // this is put under up to keep tree as flat as
               possible
            else Union(this,up)
            // this linked to up
        end
    end
    foreach element $e$ of the line $i$ do FindCompress($p$)
end
```

Algorithm 1: disjoint decomposition