```
----------------------
 Algorithm 3
----------------------
 FindVisible_3(M[1..n], B[1..n])
     n = len(M)
     Initialize each line as invisible:
     V[i] = False, i = 1..n
     V[1] = V[2] = True

     SubVis = [1, 2]  # index of current visible lines
     for i = 3, i <= n, i++:
         while len(SubVis) > 1
             j = len(SubVis)
             (x, y) = Intersection point of line j and line j-1 in SubVis
             if y < M[i]*x + B[i]:
                 Remove last line from SubVis
             else:
                 break
         Append Yi to SubVis
     for k = 1, k < len(SubVis), k++:
         V[SubVis[k]] = True
     return V
```