```go
package main

import "fmt"
const MAXSIZE = 20192020
type RecType struct {
    count int
    size int
}
type ArrType [MAXSIZE]RecType

func main(){



}
func iSort(tab *ArrType, nsize int) {
    var i int
    var t int

    i = 1
    for i < nsize{
        t = tab[i].count
        j := i-1
        for j>=0 && tab[j].count > t {
            tab[j+1].count = tab[j].count
            i++
        }
    }
    tab[j+1].count = t
    i++
}

func mSort(tab *ArrType, nsize int) {
    i := 0
    for i< nsize {
        max := 1
        j := 1
        for j<nsize {
            if tab[j].size > tab[max].size {
            max = j
            }
            j++
        }
```

```go
        j := i-1
        for j>=0 && tab[j].count > t {
            tab[j+1].count = tab[j].count
            i++
        }
    }
    tab[j+1].count = t
    i++
}

func mSort(tab *ArrType, nsize int) {
    i := 0
    for i< nsize {
        max := 1
        j := 1
        for j<nsize {
            if tab[j].size > tab[max].size {
            max = j
            }
            j++
        }
        temp := tab[i].size
        tab[i].size = tab[max].size
        tab[max].size = temp
    i++
    }
}

func isFound(tab ArrType, n int, v int) bool {
    kr := 0
    kn := n
    found := false
    for kr < kn && !found {
        med := (kr+kn)/2
        if tab[med].count < v {
            kr = med + 1
        }else if tab[med].count > v {
            kn = med
        }else {
            found = true
        }
    }
    return found
}
```