```
package org.training.spark.testing

import java.util._

import scala.collection.JavaConversions._

/**

 * Created by hduser on 10/5/17.

 */

object docks {


 var sol: ArrayList[Integer] = new ArrayList[Integer]()

 def main(args: Array[String]): Unit = {

  val sc: Scanner = new Scanner(System.in).useDelimiter("\\n")

  var t: Int = sc.nextInt()

  while ({ t -= 1; t + 1 } > 0) got.dockcalculator()

  for (a <- sol) println(a)

 }

 def dockcalculator(): Unit = {

  val sc: Scanner = new Scanner(System.in).useDelimiter("\\n")

  var c: Int = 0

  var part: Array[String] = Array.ofDim[String](2)

  c = sc.nextInt()

  val a: Array[Int] = Array.ofDim[Int](c)

  val d: Array[Int] = Array.ofDim[Int](c)
```

```
for (i <- 0 until c) {
  val s: String = sc.next()
  part = s.split(" ")
  a(i) = java.lang.Integer.parseInt(part(0))
  d(i) = java.lang.Integer.parseInt(part(1))
}
var temp: Int = 0
for (k <- 0 until c; l <- 0 until c - 1 if a(l + 1) < a(l)) {
  temp = a(l)
  a(l) = a(l + 1)
  a(l + 1) = temp
  temp = d(l)
  d(l) = d(l + 1)
  d(l + 1) = temp
}
var max: Int = 0
var counter: Int = 1
var i: Int = 0
var x: Int = 0
for (k <- 1 until c) {
  while (max < k) {
    if (a(k) == d(i)) {
```

```
      { max += 1; max - 1 }
      { i += 1; i - 1 }
      //break
    }
    if (a(k) < d(i)) {
      { counter += 1; counter - 1 }
      { max += 1; max - 1 }
    }
    x = 0
    if (a(k) > d(i)) {
      { i += 1; i - 1 }
      { x += 1; x - 1 }
      { max += 1; max - 1 }
    }
    if (d(i) > a(c - 1)) {
      { i += 1; i - 1 }
    }
    if (x != 0) {
      counter = counter - x + 1
    }
  }
}
```

```
    sol.add(counter)
  }


}
```