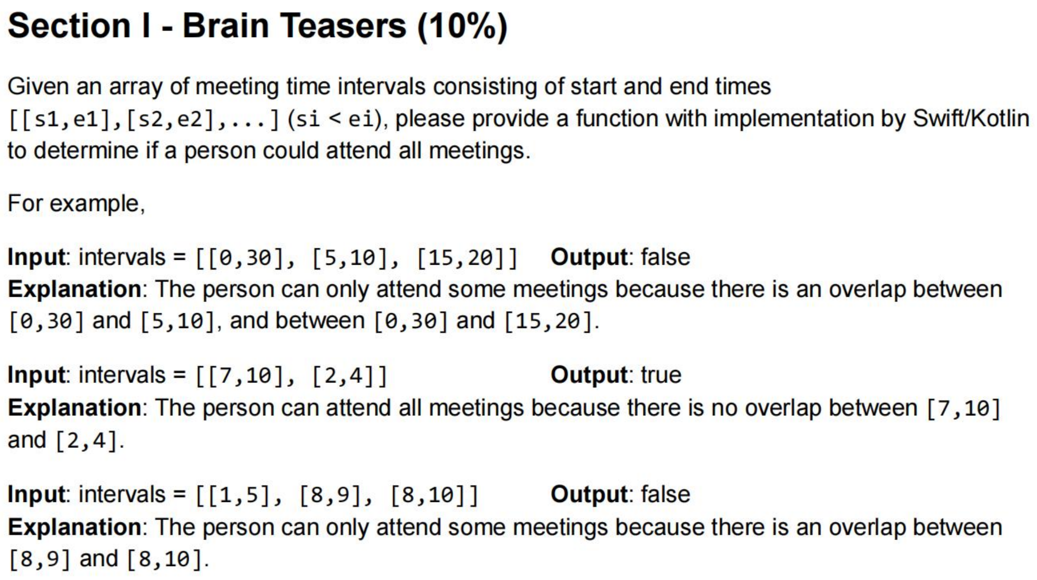

Section I - Brain Teasers (10%)

Given an array of meeting time intervals consisting of start and end times `[[s1,e1],[s2,e2],...]` (`si` < `ei`), please provide a function with implementation by Swift/Kotlin to determine if a person could attend all meetings.

For example,

Input: intervals = `[[0,30], [5,10], [15,20]]`　**Output**: false
Explanation: The person can only attend some meetings because there is an overlap between `[0,30]` and `[5,10]`, and between `[0,30]` and `[15,20]`.

Input: intervals = `[[7,10], [2,4]]`　**Output**: true
Explanation: The person can attend all meetings because there is no overlap between `[7,10]` and `[2,4]`.

Input: intervals = `[[1,5], [8,9], [8,10]]`　**Output**: false
Explanation: The person can only attend some meetings because there is an overlap between `[8,9]` and `[8,10]`.

```
var meetingTime1: List<List<Int>> = listOf(listOf(5, 10), listOf(15, 20), listOf(0,
30))
var meetingTime2: List<List<Int>> = listOf(listOf(7, 10), listOf(2, 4))
var meetingTime3: List<List<Int>> = listOf(listOf(1, 5), listOf(8, 10), listOf(8,
9))

fun canAttendMeetings(meetingTime: List<List<Int>>): Boolean {
    //按照开始时间对会议进行排序（即小到大排序）
    var tempMeetingTime = meetingTime.sortedBy { it.first() }
    //遍历会议,检查每个会议结束时间是否在下个会议的开始时间之内
    for (i in tempMeetingTime.indices) {
        if (i < tempMeetingTime.size - 1) {
            if (tempMeetingTime[i][1] > tempMeetingTime[i + 1][0]) {
                return false
            }
        }
    }
    return true
}
```