Merge Intervals

Question: Given a collection of intervals, merge all overlapping intervals.

For example:

Given [1,3],[2,6],[8,10],[15,18],

return [1,6],[8,10],[15,18].

Solutions:

```
class Interval:
    def __init__(self, s=0, e=0):
        self.start = s
        self.end = e

    def println(self,i):
        print ("[%d ,%d]"%(i.start,i.end))

class Solution:
    # @param intervals, a list of Interval
    # @return a list of Interval
    def merge(self, intervals):
        intervals.sort(key = lambda x:x.start)
        length=len(intervals)
        res=[]
        for i in range(length):
            if res==[]:
                res.append(intervals[i])
            else:
                size=len(res)
                if res[size-1].start<=intervals[i].start<=res[size-1].end:
                    res[size-1].end=max(intervals[i].end, res[size-1].end)
                else:
                    res.append(intervals[i])
        return res

i1 = Interval(1,3)
i2 = Interval(2,6)
i3 = Interval(8,10)
i4 = Interval(15,18)

result = Solution().merge([i1,i2,i3,i4])
for i in result:
    Interval().printIn(i)
```