```
class Solution {
   static class Meeting {
        int startTime;
        int endTime;

        public Meeting(int startTime, int endTime) {
            this.startTime = startTime;
            this.endTime = endTime;
        }

    }

    public int maxMeetings(int start[], int end[]) {
        List<Meeting> meetings = new ArrayList<>();
        for (int i = 0; i < start.length; i++) {
            meetings.add(new Meeting(start[i], end[i]));
        }

        Collections.sort(meetings, new Comparator<Meeting>() {
            @Override
            public int compare(Meeting o1, Meeting o2) {
                return o1.endTime - o2.endTime;
            }
        });

        int maxi = 1;
        Meeting currMeeting = meetings.get(0);
        for (int i = 1; i < meetings.size(); i++) {
            if (currMeeting.endTime < meetings.get(i).startTime) {
                maxi++;
                currMeeting = meetings.get(i);
            }
        }
        return maxi;
    }
}
```