**Steps we followed to make it work:**

1. we need to first manufacture and then only it can go for flight testing.
2. hence, we sorted all the intervals based on manufacturing time which took O(nlogn) of time for sorting.
3. Now, we know which task we need to pick first.
4. Next, we need to schedule flight testing once manufacturing is completed.
5. But we need to keep in mind if there is a gap in manufacturing time and last flight testing, then the flight testing scheduler is in IDLE state, which we need to count.
6. Hence we run a loop to check if the current manufacturing time is more than or equal to the last flight testing and if we put IDLE state in the flight Testing schedule.
7. so we need O(n) for running for all the intervals of time
8. Total complexity of the code is O(n) + O(nlogn) which is O(nlogn)

**Class used for making it work:**

**class Interval:**

it's a POJO class for interval where we have id, flight testing time and manufacturing time

**class Drone:**

it's the main class which does everything for scheduling the task, we have a method which takes the input and prepares the interval list.