Python Task — Intervals Toolkit

Implement four small functions that operate on half-open time intervals: `(start, end)` where `0 ≤ start < end ≤ 1440`, measured in minutes from the start of day.
Touching intervals (e.g., `(10,20)` and `(20,30)`) do NOT overlap. No third-party libraries needed; standard Python only.

What to build

```
1) merge_intervals(intervals) -> list[tuple[int,int]]
   Return intervals merged so none overlap. Keep touching intervals separate.

2) total_occupied_time(intervals) -> int
   Total minutes covered by the intervals without double counting overlaps.

3) can_schedule(intervals, new_interval) -> bool
   True if new_interval doesn't overlap any in intervals (touching is allowed).

4) max_overlap(intervals) -> int
   Maximum number of intervals active at the same time.
```

Examples

```
intervals = [(30, 90), (0, 15), (60, 120), (120, 180), (90, 100), (200, 240)]
merge_intervals(intervals)
# -> [(0, 15), (30, 120), (120, 180), (200, 240)]

total_occupied_time(intervals)
# -> 15 + 90 + 60 + 40 = 205

can_schedule(intervals, (15, 30))  # touches only
# -> True

can_schedule(intervals, (100, 130))
# -> False

max_overlap(intervals)
# -> 3
```

Quick sanity checks

```
if __name__ == "__main__":
    intervals:list[tuple[int, int]] = [(30, 90), (0, 15), (60, 120), (120, 180), (90, 100), (200, 240)]
    assert merge_intervals(intervals) == [(0, 15), (30, 120), (120, 180), (200, 240)]
    assert total_occupied_time(intervals) == 205
    assert can_schedule(intervals, (15, 30)) is True
    assert can_schedule(intervals, (100, 130)) is False
    assert max_overlap(intervals) == 2
    print("All good!")
```