Daily Coding Problem #171

Problem

This problem was asked by Amazon.

You are given a list of data entries that represent entries and exits of groups of people into a building. An entry looks like this:

`{"timestamp": 1526579928, count: 3, "type": "enter"}`

This means 3 people entered the building. An exit looks like this:

`{"timestamp": 1526580382, count: 2, "type": "exit"}`

This means that 2 people exited the building. `timestamp` is in Unix time.

Find the busiest period in the building, that is, the time with the most people in the building. Return it as a pair of `(start, end)` timestamps. You can assume the building always starts off and ends up empty, i.e. with 0 people inside.

Solution

We can solve this by first sorting the entries by the timestamp (assuming they're not already sorted). Then, as we iterate over the sorted entries we can keep track of the current number of people. If we go over the max we've seen so far, then we update the max number of people and the bounds of the period.

```python
def busiest_period(entries):
    period = (None, None)
    num_people, max_num_people = 0, 0

    # Sort the entries by timestamp
    sorted_entries = sorted(entries, key=lambda e: e["timestamp"])

    # Keep track of the number of people at each entry.
    for i, entry in enumerate(sorted_entries):
        if entry["type"] == "enter":
            num_people += entry["count"]
        else:
            num_people -= entry["count"]

        if num_people > max_num_people:
            max_num_people = num_people
            period = (entry["timestamp"], sorted_entries[i + 1]["timestamp"])

    return period
```

Since we have to sort the entries this takes O(n log n) time. If they're already sorted, then this just takes O(n) time.