Interview Cake

← course home (/table-of-contents#section_sorting-searching-logarithms_question_merging-ranges)

Your company built an in-house calendar tool called HiCal. You want to add a feature to see the times in a day when *everyone* is available.

To do this, you'll need to know when *any* team is having a meeting. In HiCal, a meeting is stored as a tuple of integers `(start_time, end_time)`. These integers represent the number of 30-minute blocks past 9:00am.

For example:

```
(2, 3)  # Meeting from 10:00 - 10:30 am
(6, 9)  # Meeting from 12:00 - 1:30 pm
```

Python 2.7

Write a function `merge_ranges()` that takes a list of multiple meeting time ranges and returns a list of condensed ranges.

For example, given:

```
[(0, 1), (3, 5), (4, 8), (10, 12), (9, 10)]
```

Python 2.7

your function would return:

```
[(0, 1), (3, 8), (9, 12)]
```

Python 2.7

Do not assume the meetings are in order. The meeting times are coming from multiple teams.

Write a solution that's efficient even when we can't put a nice upper bound on the numbers representing our time ranges. Here we've simplified our times down to the number of 30-minute slots past 9:00 am. But we want the function to work even for very large numbers, like Unix timestamps. In any case, the spirit of the challenge is to merge meetings where `start_time` and `end_time` don't have an upper bound.
