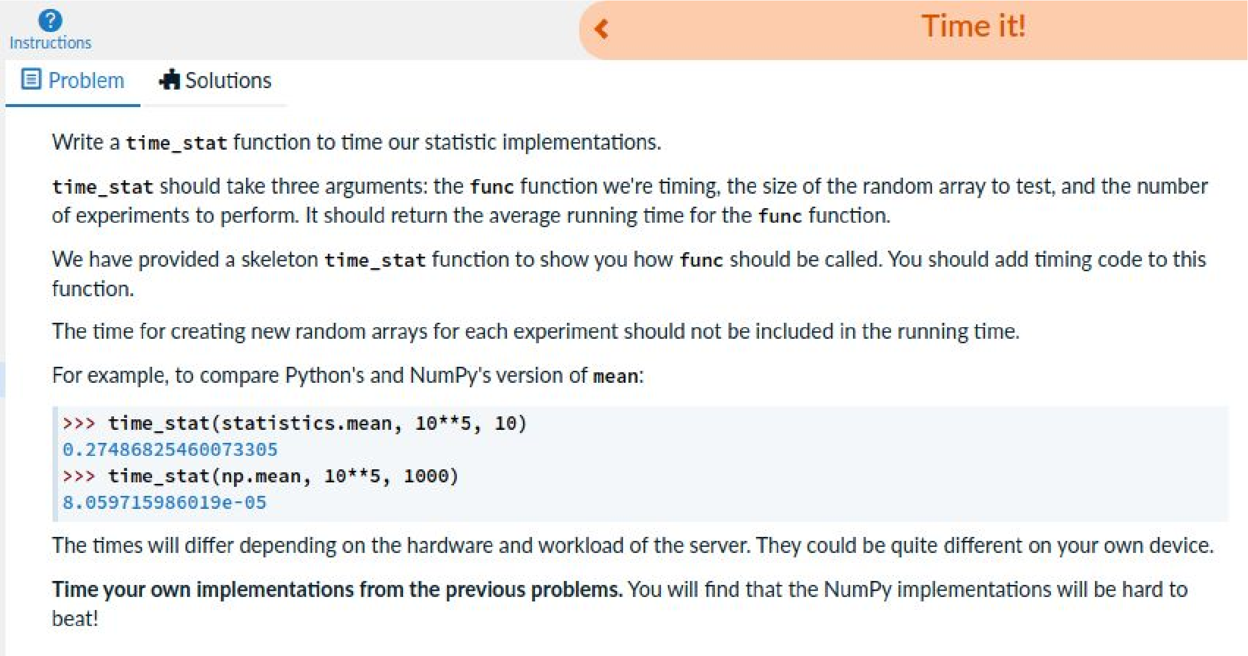

Time it!

Write a `time_stat` function to time our statistic implementations.

`time_stat` should take three arguments: the `func` function we're timing, the size of the random array to test, and the number of experiments to perform. It should return the average running time for the `func` function.

We have provided a skeleton `time_stat` function to show you how `func` should be called. You should add timing code to this function.

The time for creating new random arrays for each experiment should not be included in the running time.

For example, to compare Python's and NumPy's version of `mean`:

```
>>> time_stat(statistics.mean, 10**5, 10)
0.27486825460073305
>>> time_stat(np.mean, 10**5, 1000)
8.059715986019e-05
```

The times will differ depending on the hardware and workload of the server. They could be quite different on your own device.

Time your own implementations from the previous problems. You will find that the NumPy implementations will be hard to beat!