Simulated coursework exercise - Quicksort

This is a commonly used searching algorithm that works by divide-and-conquer.
Given some sort of comparable input, for example a sequence of integers, you split it up according to a certain 'pivot' value, and make lists of values that are less, equal or greater than this pivot.
You then recursively split each list up.
Once you have fully split up the list, you recombine them in the order less, equal and greater and the resulting list will be fully sorted.
More information on the quick sort is available here:
https://en.wikipedia.org/wiki/Quicksort

Your task is to **implement a quicksort** method using the python language. You must use version **python3.4 or higher** for this task.
Your solution should implement the following interfaces:

A file named "quicksort.py":

- Contains a method "quicksort":
  - that takes in a list of integers, possibly unsorted
  - And returns a sorted version, according to the quicksort

Some test files. These should follow the standard unit testing framework offered by python 3. You should use these test files to check your own solutions, but remember that they will be used on other people's solutions:

- They should assume that the file "quicksort.py" is in the same directory.
- They are only required to test what is offered by the interface above.

Within the coursework, you will also find some skeleton files that offer the interfaces, for you to fill in yourself.