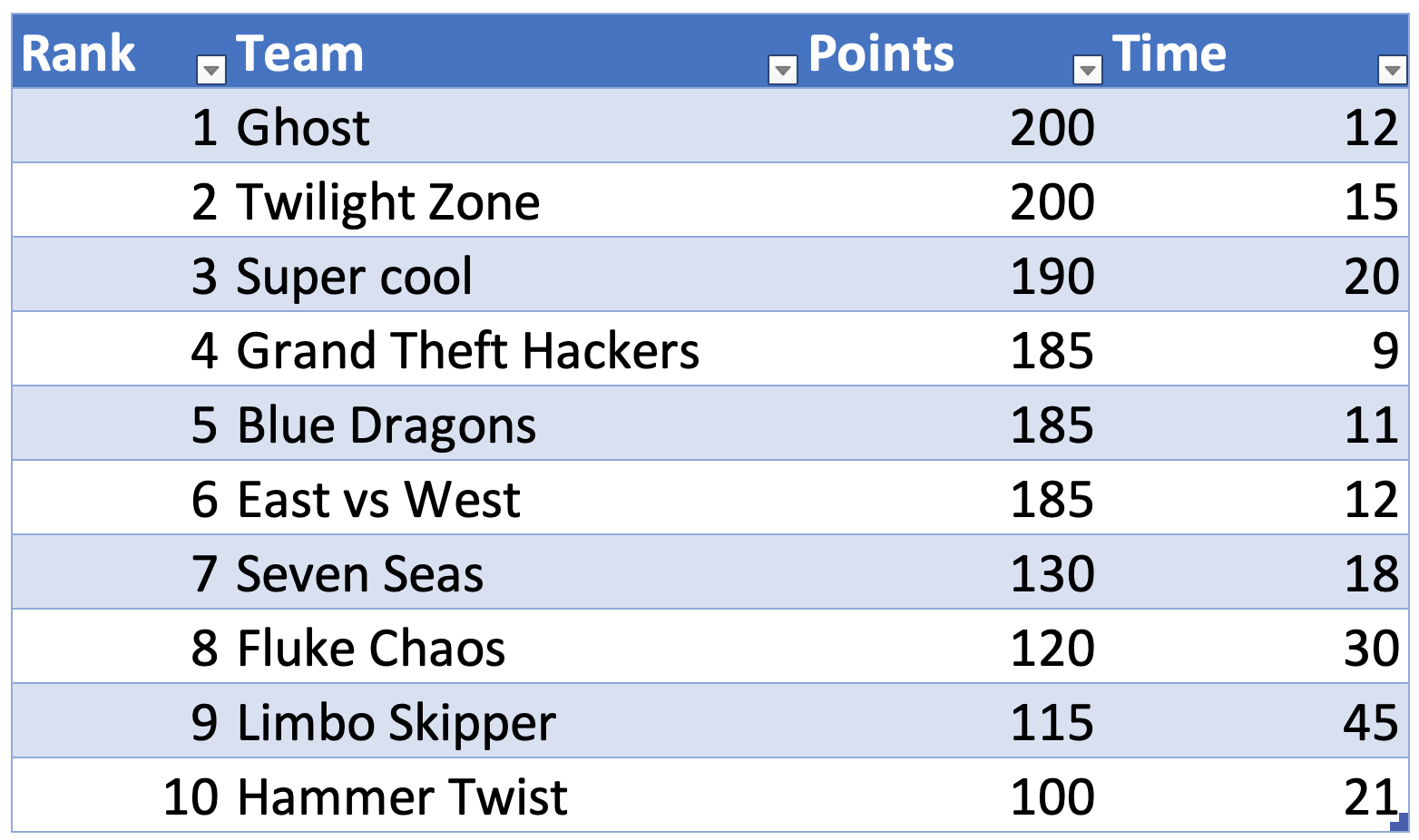
Lab 6b – Scoreboard ADT

In a particular programming competition, teams create programs to solve problems and are awarded points. At the end of the competition the team with the highest number of points wins. Because some teams can end up with the same number of points the competition will also use the time (measured in minutes) so that teams who solve the problem completely is less time will end up at the top of the board. An example scoreboard is shown here:



To implement the display of this scoreboard, the software architect has called for a Python class called “ScoreBoard” that supports the following operations:

* addScore(team, points, time)
* clear()
* \_\_iter\_\_()
* \_\_next\_\_()

The addScore function will add a new score to the scoreboard. The scoreboard will only retain the top 10 scores so it is possible that if you have 10 scores already and you try to insert a new result with a low score it could be just ignored (because it is not one of the top 10).

If a score is entered for a team that already exists, the add\_score() function must raise an exception.

If there are two teams with the same points and time, then the order will be done alphabetically with teams starting with the letter A appearing before teams starting with the letter B.

The class is to be made iterable so that the other part of the program using the class can use the “for x in” style of iteration. The next() operator should return the next item in the list as a dictionary of the form: { “team”: “Ghost”, “points”: 200, “time”: 200 }.

Write appropriate unittest cases for your data structure.