Shuffled Exams

Here is a video walkthrough of the solutions.

For this problem, we will be working with `Exam` and `Student` objects, both of which have only one attribute: `sid`, which is a number like any student ID.

PrairieLearn thought it was ready for the final. It had meticulously created two arrays, one of `Exams` and the other of `Students`, and ordered both on `sid` such that the ith `Exam` in the `Exams` array has the same `sid` as the ith `Student` in the `Students` array. Note the arrays are not necessarily sorted by `sid`. However, PrairieLearn crashed, and the `Students` array was shuffled, but the `Exams` array somehow remained untouched.

Time is precious, so you must design a O(N) time algorithm to reorder the Students array appropriately **without** changing the `Exams` array!

Hint: Begin by reordering **both** the `Students` and `Exams` arrays such that ith `Exam` in the `Exams` array has the same `sid` as the ith `Student` in the `Students` array.

Solution:
Let's begin by creating an `ExamWrapper` class that contains two attributes — an `Exam` instance and the `index` of the corresponding `Exam` in the `Exams` array. Next, for each `Exam`, create the corresponding `ExamWrapper` instance.

Run radix sort on the `ExamWrappers`, sorting them on the `sid` of the `Exam` instances. Similarly run radix sort on the list of `Students`, sorting them on `sid` as well. Note that both iterations of radix sort take linear time since the `sid` is of fixed length and of base 10.

At this point in the algorithm, we have "completed" the hint, but we still need move the ith `Student` to its proper place relative to the original `Exams` array. To acheive this, for the ith `Student`, we will access the ith `ExamWrapper`, and set the index of the ith `Student` as the `ExamWrapper's index` attribute.

.