Eric Roberts
CSCI 121

Handout #40
October 26, 2018

CSCI 121 Sorting Contest

Submission deadline: Monday, November 5, 11:59 P.M.

Of the three contests in CSCS 121, the sorting contest is the easiest to describe. Your job is to write a Python implementation of the function **`sort`** with the following prototype:

```
def sort(array):
```

The function is called with an array of values of type **`int`** and returns after sorting the elements of that array in nondecreasing order. The entries will be judged entirely by how much time they require to sort various arrays of 1,000,000 integers, although you have no way of knowing exactly what those arrays will be. Although Python allows integers to grow to arbitrary sizes, the array passed to **`sort`** will contain only integers that fit in 32 bits, which means that their range extends from −2,147,483,648 to 2,147,483,647.

This problem is all too simple if you can use predefined methods like **`sort`** or the built-in functions **`min`**, **`max`**, or **`sorted`**, so it is important to restrict the problem so that the only array operations you can perform are:

- Using **`len`** to determine the length of an array
- Passing arrays to functions as arguments or returning them as results
- Creating an array of length **`n`** using the repetition construct **`[0] * n`**
- Selecting a single element using square brackets
- Selecting a slice (as the textbook implementation of merge sort does)

Prizes

As on the graphics contest, the prize for winning the sorting contest is that we will replace whatever individual score most negatively affects your grade—which may be an assignment, the midterm, or the final—with a 100% in the computation of the final grade. As announced in class, anyone who submits an entry that follows the rules gets a ticket for the random prize drawing at the end of the semester.

Official rules

1. Only students registered in CSCI 121 are eligible to submit entries in the contest.
2. Only one entry per person will be accepted.
3. All entries must be submitted by 11:59 P.M. on Monday, November 5.
4. To be eligible, your program must correctly sort the numbers (it's surprising how many buggy sorts have been submitted at Stanford over the years) and must not use any array operations other than the ones specifically permitted.
5. You may look up sorting algorithms in library texts and use those algorithms as models. If you do undertake such research, remember to cite your sources in the program comments. Do not, however, ask for help from members of the course staff.