COP 3502 Summer 20 Section 2 Recitation Program #3

A Prize No One Can Win

For each recitation program, in order to get full credit, you must submit your solution to open.kattis.com and get your solution accepted on all test cases. In addition, each one will have some separate requirements to submit. When submitting your work to Webcourses, please carefully read the corresponding directions document before submitting all of your files.

NOTE: Over the course of the semester, you must submit TWO out of the four recitation programs. It is expected that while you are in recitation, you start working on each of them. But, afterwards, you can choose which two to finish up.

What This Program Is Testing

The "hardest" part of this problem is sorting the input data efficiently. Since there are upto 10^5 input values to sort, an $O(n^2)$ sort will receive a Time Limit Exceeded verdict. Thus, it's important to use an O(nlgn) algorithm to sort the input list. Once the input data is sorted, a minor observation needs to be made and a single for loop through the sorted data will solve the problem.

Most languages have built in sorting functions, but the whole point of this portion of the course is manually writing your own sorting functions, and also being able to debug sorting functions.

For this assignment, please COPY the implementation of quick sort on the course web page:

http://www.cs.ucf.edu/~dmarino/ucf/transparency/cop3502/sampleprogs/quicksort.c

Then, write code to solve the problem using this sorting function to sort the data.

Then, submit, your code. You should get **"Time Limit Exceeded"**, due to the sort function running too slowly.

Look over the code carefully, and try to figure out WHY it receives time limit exceeded. Then, edit the code to speed it up. If you do this properly, and use the sorted array correctly to solve the problem, then you should get an **"Accepted"** verdict.

Thus, I am specifically testing two things here: (a) Your ability to solve the problem given a sorted array, coming up with your own original solution, (b) Your ability to find the reason why my posted quick sort code runs very slowly in some cases and to fix it to run in O(nlgn) in a majority of cases.

What to Submit

Please submit the following:

1) Your source file, prize.c, that was Accepted.

2) A screenshot of your solution's "Time Limit Exceeded" status on Kattis.

3) A screenshot of your solution’s "Accepted" status on Kattis.

4) A document (.txt, .doc or .docx) explaining the edit you made to my posted quicksort.c code, for which input arrays your edit sped up the code, and why that speed up was significant.