Illinois Institute of Technology
Department of Computer Science

Homework Assignment 0

CS 535 Design and Analysis of Algorithms
Fall Semester, 2015

Rules for Homework

On this and all future homework assignments, observe these guidelines *or credit will be deducted*:

- Write your name *clearly* at the top of *each* page. If the TA and I cannot read your name *easily*, you will not get credit for the assignment.
- Begin each problem on a new page.
- Use standard 8.5 inch by 11 inch paper.
- Include the Honesty Pledge. No homework will be graded until a signed pledge has been submitted.
- Staple your finished assignment in the upper-left corner. Do not use a paper clip or fold the corner—it is your job to insure that pieces of your assignment don't get lost.
- You may discuss only general problem-solving strategies with other students; your homework solutions *must be entirely your own work* and clearly distinguished from other homeworks. If there is any evidence of collaboration, or seeking/offering/obtaining any outside help, *you will fail the course*—no exceptions; sadly, this has happened, so don't let it happen to you. Any resources (aside from the text and class notes) in print or on the web, that you use in solving the homework problems must be explicitly cited (give full citations—authors, titles, dates of publication, page numbers, web sites). You must sign and attach the *Honesty Pledge* with this first assignment.
- Algorithms must be carefully presented, proved correct, and analyzed in terms of the their time and space requirements. Follow the style of the textbook in these matters—your analyses and proofs must be mathematically rigorous.

Due: Thursday, August 27, 2015

1. CLRS3 has code for Randomized-Selection on page 216. Unfortunately, a CS 535 student implementing the code mistyped q instead of $q-1$ in line 8.

   (a) Does the corrupted code still work (that is, correctly find the ith smallest element) always, sometimes, or never? Explain.

   (b) Analyze the worst-case running time of the corrupted code.

   (c) Analyze the best-case running time of the corrupted code.

   (d) Analyze the average-case running time of the corrupted code. (*Hint*: Be careful that you do not get snagged by the pitfall described in the middle of page 86 of CLRS3.)

   (e) There is (should be, if you did it correctly) something strange about your answer to the previous part; what is strange and how do you explain it?