Homework 1

Due February 7

This is the last homework on which the following information will appear, but remember that it applies to every homework:

- Homeworks are due at 11:59 PM on the date specified. Homeworks are to be placed in the CS 16 handin bin, located by the glass doors on the second floor.
- Please staple your homeworks and put your name and login on each page.
- Please ensure that when prompted to provide pseudocode you follow proper pseudocode formatting guidelines. See the relevant links in the Information section of the website. Commenting of your pseudocode is *strongly* encouraged.
- If you use LaTeX, use the `newalg` or `algorithmic` packages (`http://www.cs.brown.edu/system/software/latex/packages.html`) to format your pseudocode.
- When writing pseudocode, you may use any algorithms in the lecture slides or course texts for which there is pseudocode provided. If you do so, please cite the specific lecture slide or page from the book where the algorithm is described.
- Credit for problems comes in part from the simplicity of your answers; insanely long answers lose credit.
- Use pictures to illustrate your ideas.
- Write neatly. Hard-to-read homework gets no credit. If you scratch things out, rewrite and hand in a clean copy.

This is a collaborative howework.

Problem 1.1

Suppose an initially empty stack S has performed a total of 25 `push` operations, 12 `top` operations, and 10 `pop` operations. Three of the `pop` operations generated EmptyStackExceptions, which were caught and ignored. Is this enough information to determine the current size of S? If so, what is it? If not, why not?

Problem 1.2

Suppose you have a stack S containing n elements and a queue Q that is initially empty. Describe how you can use Q to scan S to see if it contains a certain

element x, with the constraint that your algorithm must finish with S in the same state that it started in, i.e., it must contain the same elements in the same order. You may not use any other data structures – only S and Q and a constant number of reference variables.

Problem 1.3

(a) Describe a recursive algorithm for enumerating all permutations of the numbers $\{1, 2, \ldots, n\}$.

(b) Describe a non-recursive algorithm for enumerating all permutations of the numbers $\{1, 2, \ldots, n\}$.

Problem 1.4

Forty-eight notation is an unambiguous way of writing an arithmetic expression without parentheses. (For simplicity, we will consider only arithmetic expressions consisting of integers and the binary operations +,-,*, and /.) It is defined so that if "(exp_1)**op**(exp_2)" is a normal fully parenthesized expression whose operation is *op*, then the forty-eight version of this is "$pexp_1$ $pexp_2$ **op**", where $pexp_1$ is the forty-eight version of exp_1 and $pexp_2$ is the forty-eight version of exp_2. The forty-eight version of a single number is just the number itself. So, for example, the forty-eight version of "$((5+2)*(8-3))/4$" is "$5 \quad 2+8\ 3-*\ 4\ /$". Describe a non-recursive algorithm for evaluating an expression in forty-eight notation. Hint: there is a data structure that will be particularly useful.

Problem 1.5

When a share of common stock of some company is sold, the capital gain (or loss) is the difference between the share's selling price and the price originally paid to buy it. This rule is easy to understand for a single share, but if we sell multiple shares of stock bought over a long period of time, then we must identify the shares actually being sold. A standard accounting principle in such a case is to use a FIFO protocol–the shares sold are the ones that have been held the longest. For example, suppose we buy 100 shares at \$20 each on day 1, 20 shares at \$24 on day 2, 200 shares at \$36 on day 3, and then sell 150 shares on day 4 at \$30 each. The capital gain in this case would be $(100*10)+(20*6)+(30*(-6))$, or \$940. Give the pseudocode for a program that takes as input a sequence of transactions of the form "buy x shares at \$$y$ each" or "sell x shares at \$$y$ each," assuming that the transactions occur on consecutive days and the values x and y are integers. Given this input sequence, the output should be the total capital gain (or loss) for the entire sequence, using the FIFO protocol to identify shares.