

CSCI2100: Special Exercise Set 6

Prepared by Yufei Tao

Problem 1. Describe the output of each `pop()` operation in the following sequence of operations on an initially empty stack:

`push(56), push(6), push(83), pop(), push(15), pop(), pop()`.

Problem 2. Describe the output of each `de-queue()` operation in the following sequence of operations on an initially empty stack:

`en-queue(56), en-queue(6), en-queue(83), de-queue(), en-queue(15), de-queue(), de-queue()`.

Problem 3. Consider a sequence of n brackets, where each bracket is either opening (namely “[”) or closing (“]”). The sequence is *legal* if, intuitively, every opening bracket finds its closing counterpart. For example, `[] []` is legal but `[]]` is not. Formally, a legal sequence is such that, one can continuously remove two adjacent brackets `[]` until all the brackets have disappeared. Suppose that the sequence is stored in an array of length n , where each bracket is stored in a cell. Give an algorithm to check whether the sequence is legal in $O(n)$ time.

Problem 4 (Elite). Consider the algorithm for the “stack-with-array” problem, namely, the algorithm that implements a stack as a dynamic array. Suppose we perform 30 pushes, followed by 20 pops, and then by another 20 pushes. What is the length of the array? Remember that the length of the array refers to the specified length when the array is created; it does not mean the number of elements stored in the array.

Problem 5. Let A be an array of n integers, each of which comes from the domain $[1, U]$. Note that some of the integers may be identical. Design an algorithm to compute, for each distinct integer x in A , how many integers in A are at most x . For example, if A stores the sequence of integers `(35, 12, 28, 12, 35, 7, 63, 35)`, you should output an array `((7, 1), (12, 3), (28, 4), (35, 7), (63, 8))`. Your algorithm should terminate in $O(n \log n)$ time.