

# CSci 4041, Spring 2007: Programming Assignment 1

Due electronically on Friday 2/16 at NOON.

All programming assignments are due at or before **noon** on the date indicated. This deadline is **STRICT**. Please be sure to leave yourself sufficient time to port your program to an ITlabs machine if you have not been working on one already. Also be sure to leave enough time to write the README documentation and to package the project into a tar.gz file. Assignments can be completed in either C/C++ or Java.

1. [20 pts.] **Ten Best List:** You are part of a team implementing a program that will be used for judging the prize winners in an intergalactic bass-fishing tournament. Your job is to create a data structure for keeping track of the 10 best fish (in terms of length - a longer fish is a better fish). Every time a new fish (length) is submitted, your data structure should add it to the sorted 10 best list if it is better than any fish in that list, and remove the shortest fish that was in the list. You can assume that all fish lengths are integers.

Since the tournament is intergalactic, and there are potentially infinite galaxies, there are potentially an unlimited number of submissions, so your data structure should be able to scale so that it can receive an arbitrary number of submissions (i.e. you shouldn't store every single submission). Your program should be as time and space efficient as possible on arbitrarily large inputs.

More formally, you need to implement and test a data structure which takes as input an arbitrarily long and unsorted list, which is transmitted one element at a time, and returns the sorted 10 highest values of the input. Thus, your data structure should have an interface to the world which consists of a method for submitting a fish, and a method for output. See below for a toy example of how your data structure will be used. You also need to implement a testing program for your data structure which will take as an argument a filename, and read an integer from each line of the file which will represent one submitted fish length.

2. [10 pts.]

**Correctness proof:** Based on whatever structure you choose for your data structure, use loop invariants or recursion invariants and an inductive proof showing the correctness of your algorithm for solving this problem.

3. [10 pts.]

**Time-complexity analysis:** Perform a run-time analysis that results in a fairly tight big-O upper bound on the run-time of your algorithm.

As an example, given the list

[4 1 3 6 20 33 15 6 2 77 10 12 8 88]

the resulting output should be:

88 77 33 20 15 12 10 8 6 6

We will post a sample input file here shortly, along with a text file containing the desired corresponding output. Your program should produce a text file indistinguishable from the supplied output file. For submission instructions, see the instructions on the course website for submitting programming assignments.