

CSE331 – Project 1: Insertion Sort

In this project you will complete an implementation of the insertion sort algorithm. The project must be submitted to Handin no later than 11:59 pm Wednesday January 16, 2013. Email the TAs with any questions. Good Luck!

Project Description

You are being provided with sample code and test input for a program that reads a data file containing the filmographies for actors in the Internet Movie Database. The program takes two command line arguments: (1) the number of actors to be read in and (2) the datafile to read from. The program reads in the movies from the specified number of actors, sorts the movies in alphabetical order, outputs them, then exits.

Your assignment is to complete the implementation of insertion sort found in InsertionSort.h. Make no other changes to the files other than completing the assigned method. You must provide your own code for insertion sort, you may not copy another's code, and you may not make any other explicit function calls from the assigned method. You may not use any of the tools provided in the STL, other than those in STL vector (i.e. you may not call STL sort, or any other STL algorithm, but you may use `vector::size()` and `vector::resize()`).

The code and a small example file may be found in `~/cse331/Projects/Project1/`. The file containing the entire Internet Movie Database may be found in `~/cse331/Projects/Data/`.

Programming Notes

The `CInsertionSort` class is template, so that it may perform the sort without specific knowledge of the exact types being sorted. You may assume that any class for which this insertion sort is instantiated will provide the following operators: `=`, `==`, `<`. These are the only operators you are guaranteed to have.

Project Deliverables

The following files must be submitted via Handin no later than 11:59 pm Wednesday January 16, 2013:

- `InsertionSort.h` – contains your implementation of insertion sort
- `project1.pdf` – files containing your answers to the written questions.

Written Questions

Submit the answers to these questions in a PDF file, called `project1.pdf`, along with your source code files. The versions of MS Office and OpenOffice in the CSE labs both support exporting to PDF format.

1. Run your program for data sizes in 10 steps. Choose these sizes so you have an even distribution and the maximum running time does not exceed 10 minutes (yes, I said minutes). For the number of elements you are sorting (not the number of items), indicate:
 - a. How long did the sort take?
 - b. How many actors, and movies.

NOTE: to implement this in the code, simply change the following lines of code in `Main.cpp`:

 - Line 40: `unsigned increment = 10;`
 - Comment out lines 74 and 76.
2. When you output the sorted list of movies you may notice that there are duplicate movies in the list.
 - a. Describe an algorithm to remove all duplicate movie entries in $O(N)$ time.
 - b. Provide pseudo-code for the algorithm above.
3. Look at the driver file provided in the code, provide a short paragraph to explain what the `CMoviePointer` class does, and why we are using it instead of simply storing and sorting a vector of strings.