Homework 1 – Analysis of Algorithms

Basics of Analysis of Algorithms with Sorting as an example

Due: Wednesday, September 18th 11:59pm Pacific Time

What you will learn

- Analyzing the correctness of algorithms
- Improve grasp on complexity analysis
- Write the code for Insertion Sort algorithm

Problem 1 – 50 points

Consider a sorting algorithm that takes an array A containing n numbers. It sorts the array by repeatedly swapping an element with its neighboring element if they are in the wrong order. E.g., if A[i] > A[i+1], A[i] and A[i+1] are swapped. The same process is repeated n times or until there are no swaps required. For this algorithm, do the following:

1. Write the pseudocode (10 points)
2. Prove its correctness (20 points)
3. Find the best-case and worst-case running time of the algorithm in Θ notation (20 points)

Problem 2 – 10 points

a) Is $2^{2n} = \Theta(2^n)$, $\Omega(2^n)$, or $O(2^n)$?

b) Is $2^{n+1} = \Theta(2^n)$, $\Omega(2^n)$, or $O(2^n)$?

Briefly justify your answer. (5 points each)

Problem 3 – 20 points

Suppose you have algorithms with the running times listed below (Assuming that these are the exact number of operations performed as a function of the input size n). Suppose you have a computer that can perform 10^{12} operations per second, and you need to compute a result in at most 2 hours of computation. For each of the algorithms, what is the largest input size n for which you would be able to get the result within two hours (4 points each). Please be sure to include your steps for partial credit.

a) $200\, n^2 + 5\, n + 40000$

b) $n^3 + 3$

c) $10\, n^2 + 50000$

d) $n \log_2 n$

e) 2^{2n}

Coding problem – 20 points

In this exercise, you will write the code for Insertion Sort as a member function of the template ArrayList and demonstrate its performance on an array of objects. You are provided with two files.

The file `ArrayListT.h` contains the definition and basic implementation of an array template. You are only required to implement the member function `void insertionSort()`, which you can find close to the end of the file. You are encouraged to look through the implementation of the template to familiarize yourself with the code.

The file `InsertionSort.cpp` contains code to test the sorting algorithm. If reads the array from a file `example.txt`. You will need to also test your code with the attached test files (`test1.txt`, `test2.txt`, `test3.txt`, `test4.txt`, and `test5.txt`). Feel free to create your own test cases. In your written solutions, include any comments about implementation details that you would like to add.

Note: If you choose to work in a different programming language, please let the instructor know in advance (i.e., before Wednesday, September 11th 11:59pm) by email. Include your choice of programming language in the email, and preferably use your SJSU email or include your SJSU ID. The instructor will provide you specific instructions on what you will need to submit. The test cases will remain the same.

What to turn in

A zip file called `cmpe-130-hw1-<your SJSU ID>.zip` containing the files `cmpe-130-hw1-<your SJSU ID>.pdf`, `ArrayListT.h`, `InsertionSort.cpp`, and any other test files you used. E.g., if your SJSU ID is 111000111, and you used no other test files, your submission will be named `cmpe-130-hw1-111000111.zip` including the following files:

- `cmpe-130-hw1-111000111.pdf` containing all the written solutions
- `ArrayListT.h` containing the implementation of insertion sort
- `InsertionSort.cpp` containing your modified main file

Submission checklist – READ CAREFULLY

- All required files are included in the required format and you have included the latest version of your solutions
  - To be fair to everyone, ensure that you don't miss any files in your submission. You will be penalized according to the late policy for any such adjustments.
- All sources are cited, including the classmates you discussed with
  - Note that if your solutions look similar to other sources, this will help clarify why.
- You are submitting your own work
  - While discussions are encouraged, copying and cheating are not. Review the honesty pledge for more details.

When you submit your assignment, you automatically agree to the following statement. If you do not agree, it is your responsibility to comment why.

"I affirm that I have neither given nor received unauthorized help in completing this homework. I am not aware of others receiving such help. I have cited all the sources in the solution file."