Final

Due: Thursday, 14 June 2018, 2:00 PM as a PDF on Canvas

Sorting and Reflection

(60 pts) Sorting Analysis

Download the sorting.c code here: sorting.c It is your job to explain how the four different sorting algorithms work, why they have a specific theoretical worst time complexity (Big O), and compare and contrast the theoretical worst time complexity with experimental data.

http://classes.engr.oregonstate.edu/eecs/spring2016/cs162-001/sorting.c

Your paper must include:

- (15 pts) Function Header for each sorting algorithm (bubble_sort, insertion_sort, selection_sort, merge_sort and merge).
  /**********
  * Description:
  * Parameters:
  * Returns:
  * Pre-Conditions:
  * Post-Conditions:
  **********/

- (25 pts) Very Thorough Description of each sorting algorithm (bubble_sort, insertion_sort, selection_sort, merge_sort and merge).
  - How does the algorithm work? (Text and pictures are required. Pictures cannot be taken from the internet.)
  - What does the code do? (You must answer each question/comment in the code provided and put this in your paper!)

- (20 pts) A table with at least 15 or more test cases for timing and experimenting with the sorting algorithms:

Input Values	Sorting Algorithm	Big O Complexity	Time for N:
Empty file	Bubble Sort	O(N^2)	N=0, 0 microseconds
…			

  Compare and contrast the expected time based on the Big O complexity and data size, N, with the actual experimental time for each sorting algorithm with a data size N and the data arrangement.

  - How big did your N have to be to get timings you could use?
  - What did you data look like?
  - Do the times match what you expected?
  - Does it make any difference whether the data items are in reverse sorted order (worst case) or already sorted (best case)?

Breakdown of points/grading:

20 pts. Testing
- 10 pts. The table for 15 additional test cases (you will be graded on what you chose to make sure it worked and for timing) (2 pts for each algorithm)
- 10 pts. Compare and contrast the expected and actual complexity for each algorithm. (2 pts for each algorithm)

25 pts. Displaying understanding of the sorting algorithms
- 10 pts. Description and pictures of how the algorithm works
- 15 pts. Answers to all questions/comments in the sorting.c code.

15 pts. Function Headers (3 pts each sorting algorithm)
- Each function must have a function header included in your paper!

(40 pts) Reflection

Go to this link and take the survey:
http://oregonstate.qualtrics.com/jfe/form/SV_5tF9JaYBTrFeb2d

Answer the questions thoroughly and thoughtfully. The questions are about your experience and will not be used against you in any form. The information you provided will be used to improve the course and make adjustments based on student experience in scaling classrooms.