

COMPSCI 220 S2 C Assignment 2

Department of Computer Science

The University of Auckland

Due Sunday, 26 April 2015, 8:30 *pm*

This assignment is worth 18 marks, representing 6% of your total course grade.

Quicksort Pivots in Java

- You are given the following files, via the Cecil Resources page for COMPSCI 220, in the a2-files.zip file:
 1. a2_in.txt This is a text file containing a sequence of integers, one per line, in unsorted order.
 2. a2_quick.java An example Java program that reads integers from a file such as a2_in.txt into an integer array, then uses heapsort and quicksort (with two possible pivot functions) and prints the times taken for various array sizes.
 3. a2_out.txt The expected output for question 2 below.
- Numbers in brackets show the marks for each part of the answers.

Using automarker

We are using automarker for this assignment. Automarker is a system that runs on a remote virtual machine; you submit programs to it via its web page,
<http://www.cs.auckland.ac.nz/automated-marker>

For short-answer questions such as Question 1, submit a plain-text or .pdf file, with your answers to each section separated by blank lines. This question will be marked by human markers.

For programming questions such as Question 2, automarker takes the submitted file, compiles it and runs it. Your program should read data from standard input, and write its results to standard output.

Note that:

- Your answer to question 2 must be a single file, You may assume that automarker has access to all standard libraries.
- A sample input and output file for this question are provided. These are a2_in.txt and a2_out.txt; use them to test your program before submitting it to automarker.
- Automarker checks the output using a text comparison program, so it must be in *exactly* the right format.

CONTINUED

- You may take account of the feedback given by the automarker, and resubmit before the deadline without penalty. There is a limit of 15 submissions for each question.
- Your program(s) may be tested on much larger input files. Marks will be allocated for correctness of the programs. Simply “passing” the automarker may not guarantee maximum marks, but it will guarantee full marks for correctness.

Questions

Q1: In the lecture on quicksort (week 4) we discussed a Java implementation of quicksort; the quicksort code in `a2_quick.java` is different in several ways.

- Write brief sentences for describing a few of the differences. [4 marks]
- `a2_in.txt` contains integers in ‘almost sorted’ order; why does quicksort perform poorly on this data? [2 marks]
- Write and test a new version of function `p_index2()`, so as to significantly improve quicksort’s performance. Your answer should include your code for `p_index2()` and the output from `a2_quick` showing how well it performs. Note: `a2_quick` uses $(\text{time for quick1})/(\text{time for quick2})$ to show the ‘speedup’ of quick2 compared to quick1. [6 marks]

Q2: When you have your improved quicksort running (question 1c above), comment out (or remove) from `a2_quick.java` the statements that print the lines showing sort performance for each of the sorts.

Add some new code to print the last 20 integers in the sorted array.

Test your program using `a2_in.txt` as input and comparing your output with `a2_out.txt`. Remember, your output must match the expected output *exactly*!

Now submit your program to the automarker for question 2. If you don’t get a ‘green’ result, correct your program and submit it to the automarker again. [6 marks]

Dates and Marks

This assignment is marked out of 18 and is worth 6% of your course grade. You should submit via the automarker:

- A file containing your answers for question 1. This may be a plain text file called ‘`q1.txt`’ or a PDF file called ‘`q1.pdf`’
 - A source file for question 2. The name of this file must be ‘`a2_quick.java`’ (i.e. the same as the automarker names of the subproblems).
 - “Can I use a programming language other than Java?” Well, maybe – automarker can run python, ruby or C. However, *my testing this assignment with python showed that its interpreting overhead is much higher than that of Java, which masks the effects this assignment is designed to explore.*
-

The due date is Sunday, 26 April 2015, 8:30 pm.
