COMPSCI.220.C.S2 – Algorithms and Data Structures

Assignment 2 – Efficiency of data sorting and selection

Out: Friday, 8$^{\mathbf{th}}$ of August, 2014

Due: Friday, 22$^{\mathbf{nd}}$ of August, 2014

Assignment 2 (Lectures 7 – 12) is worth **60 marks** representing **6%** of your total course grade.

Objectives. Learning how to analyse efficiency of data sorting and selection techniques.

Requirements. You should give clear and detailed answers to the following five questions:

1. (*12 marks*) An inversion in a list $\mathbf{A}_n = [a_1, \ldots, a_n]$ of size n with different (by values) elements a_i is any ordered pair of positions (i, j) such that $i < j$ but $a_i > a_j$. Using the math induction, prove that the maximum and average numbers of inversions in such arrays are $I_{\max:n} = 0.5(n-1)n$ and $I_{\text{ave}:n} = 0.25(n-1)n$, respectively. The base case is for $n = 2$: $I_{\max:2} = 1$ and $I_{\text{ave}:2} = 0.5$.

2. (*14 marks*) You decided to modify `mergesort` by recursively splitting an array of size n into k sorted subarrays; $k > 2$, and merging these subarrays. Time for merging is $c(k-1)n$ where c is a constant factor, because at each step the largest item among the k top ones in the k subarrays is found by $k-1$ comparisons and placed into the merged array.

   Specify the recurrence relation and derive the closed-form formula for sorting time $T_k(n)$ of the modified `mergesort` for an arbitrary k (you should show all steps of the derivation). Then determine whether the modified `mergesort` could be faster for some $k > 2$ than the conventional one ($k = 2$) with the sorting time $T_2(n) = cn \log_2 n$.

   Hint: To have a well-defined recurrence, assume that $n = k^m$ with the integer $m = \log_k n$ and $T(1) = 0$. You might need also the equality $\log_2 x = \log_2 k \cdot \log_k x$ for all $x > 0$ and the inequality $k > 1 + \log_2 k$ for all $k > 2$.

3. (*12 marks*) You need to select k most successful lottery winners from an unordered list of winning records for n web lottery participants all over the world. Each record contains an ID of a participant and a winning (i.e., a value that specifies how successful this winner is). You know two strategies for selecting the k higher-rank individuals:

   (a) Run k times `quickselect` with linear processing time, $T_{\text{qselect}}(n) = cn$, in order to sequentially fetch the participants of the desired ranks $n, n-1, \ldots, n-k+1$, or

   (b) Run once `quicksort` with linearithmic processing time, $T_{\text{qsort}} = cn \log_2(n)$, to sort the list in ascending order of the winnings and fetch the k higher-rank participants.

   Let the factor c be the same for both the algorithms and let the data fetching time be negligibly small comparing to the sorting or selection time. Then find out, for which numbers k and n the option (a) is faster, and determine which option should be used if $k = 15$ and $n = 100,000,000 \approx 2^{26.6}$.

4. (*12 marks*) The first step of `quicksort` places the selected pivot to its proper sorted position i; $0 \le i \le n-1$, of the list of size n. How many inversions between the pivot in its initial position and all other $n-1$ list elements may exist in the average case and in the worst case and how many of them will be eliminated during this step?

5. (*10 marks*) A heap of size n can be built either (a) in linearithmic time, $\mathrm{O}(n \log n)$, by inserting n items one-by-one into an initially empty heap and restoring the heap property after each insertion; see Lemma 2.25 (Textbook), or (b) notably faster – in linear time, $\Theta(n)$, – by heapifying directly an entire initial list of n items; see Lemma 2.31 (Textbook). What difference between these two heap-building processes does yield such acceleration?

   Hint: Decide what property of the process (a) is redundant for building the goal heap of size n.

Your report with clearly structured and detailed answers to Questions 1–5 has to be prepared as an electronic document, e.g., by using a LaTeX document preparation system or Microsoft Word text editor, and submitted in a single PDF file with a fixed specification: `CS220assign2.pdf` (only this PDF file will be evaluated by markers, so that you must check that it can be read by PCs in the departmental Computer Labs).

Note that **scanned handwritten documents are strictly forbidden** (even as images in an electronic text) and will not be accepted for marking.

Submission

Submit your report electronically to `http://www.cs.auckland.ac.nz/automated-marker` (this automated marking system, abbreviated AMS below, will be open for submission from August 8 to August 24, 2014; 08:30 pm).

The due date is Friday, 22$^{\mathrm{nd}}$ of August 2014, 08:30 p.m. (AMS time). If submitted after due date but before 23$^{\mathrm{rd}}$ of August, 08:30 p.m., the penalty is 10%. If submitted after this date, but before 24$^{\mathrm{th}}$ of August, 08:30 p.m., the penalty is 50%; and no submission afterwards.

Marking scheme

For Questions 1 – 5	% of marks
Clear structure of your report and detailed explanations	up to 20
Correctness of your final answers	up to 20
Correctness of your intermediate steps in deriving these answers	up to 20
Detailed explanations of all steps with references, if necessary, to the textbook	up to 40
	Total: up to 100