University of Stirling

Computing Science and Mathematics

CSCU9A2 Tutorial 5 – sample answers Spring 2017

(for week starting 6 March)

1.

(a) The body of the document will use the Georgia font and text will be displayed in blue at 85% of the browser's default size for body text. If the Georgia font is not available, the text will appear in the default sans-serif font. The background of the document body will show an image, held in the file "light-blue-fade.png" in the same folder as the html and css documents. The image will be repeated horizontally across the top of the page ("repeat-x"), but will not be repeated vertically. [Note: controlling how an image repeats was not discussed in the lectures, but part of the point of these exercises is to get students to find out about web design for themselves using sources such as W3Schools.]

The first paragraph will be styled as described above for body text. In addition, the stylesheet specifies that paragraphs have no margin space above and text should be justified.

The second and third paragraphs will also be styled as described above, even though the second paragraph has class "info". This appears to be a mistake on the part of the web developer. The style sheet command div.info { color: red; } specifies a style for divisions of class "info" but does not apply to other elements of this class. The solution depends on what the developer wants to achieve. The stylesheet could be changed so that all elements of "info" class are coloured red, not just divisions. Or, the HTML could be changed to assign class "info" to the whole of the division containing paragraphs 2 and 2.

The second paragraph contains a link. The stylesheet specifies in detail how this is to be styled, depending on its status (a:link is a normal, unvisited link; visited means it has been visited; hover means the mouse is over it; active refers to the moment when it is clicked). There are some order rules: hover must come after link and visited, and active must come after hover. [Link styling was not covered in detail in the lectures – students can read about it at W3Schools.]

The final paragraph will appear in blue (why?) and will be centre-aligned because this is specified in the stylesheet for any elements with class "info".

(b) The CSS @media rule can be used for this. See `http://www.w3schools.com/css/css3_mediaqueries.asp`

2. There is no specific answer to this. It is really an opportunity to think about processes and describing them. Sensible, practical proposals for sorting processes are expected – (hopefully) described in enough detail that someone else could carry them out.

Maybe it will be a form of insertion sort or selection sort (though these are painfully slow and tedious) (a form of bubble sort is probably less likely), perhaps it will be full merge sort or quick sort (though these are probably too challenging for humans to keep track of the recursion right down to the simplest cases). Practical solutions could be to split into a number of blocks (say 20), sort each block anyhow, then merge; or to assume some kind of even distribution of student numbers and pre-sort into "buckets" (say, with common first three digits), then sort anyhow, then simply assemble the sorted blocks. *[Could avoid discussing the bucket-sort approach until the next question...]*

How the time is affected by doubling the data depends completely on the algorithm: For bubble/insertion/selection sort type algorithms, the time will go up by four times. For merge or quick sort type algorithms the time will go up by about "2.and a bit" (actually, for "ideal" versions of the algorithms (1000log1000)/(500log500) which is about 2.2). *We have not studied any of this yet up to the current point in CSCU9A2.*

3. This is hinting at the "bucket" approach above: Can reduce the big sorting problem into six smaller ones – ask students leaving the hall to put their booklet into a box depending on the first three digits of their student number (230 or less, 231, 232, 241, 242, 243 or more). Then use a basic algorithm such a selection sort within each box (if the pile is small enough to make it feasible), or indeed any algorithm at all. For this to be effective we need to have made a good choice of "bucket ranges" to divide the overall collection evenly. The total time turns out to be rather smaller than if, say, selection sort were used for the whole collection (because now the values in separate buckets are not compared with each other, whereas they are, in effect, in selection sort). Could even use the bucket approach *within* each bucket – potentially full recursive bucket sort.

4. Again, an opportunity to discuss possibilities: The obvious possibilities are: linear search (one at a time), linear search (by chunks, then work backwards when have gone too far), binary search (split at middle), and a variant of binary search where the split is made at a position guessed by looking at the student number to be found. The binary searches might be rigorously followed until the booklet is found, or maybe switched to a linear search after one or two splits. The most likely process *might* be one or two binary splits at a guessed location followed by linear search. Averages: simple linear search – 250 booklets checked on average; full binary search (middle split) – 8 or 9 booklets on average.

SBJ/DEC February 2017