Foundations of Computer Science (Semester 2) – 2015

Assessed Exercise Sheet 7 – 10% of Continuous Assessment Mark

Deadline : 11pm Sunday 8th March, via Canvas

Question 1 (18 marks)

Show how the array [5, 3, 4, 6, 8, 4, 1, 9, 7, 1, 2] would be sorted if the Heapsort algorithm was used. Describe the various processes involved and show the array as a Heap Tree at each stage.

Question 2 (20 marks)

Outline, in no more than 100 words, the general Quicksort procedure for sorting an array. [Hint: This is another example of the kind of "bookwork" question you can expect in the exam, though you will not normally be given a word limit. The answers are easily found in the lecture notes, but it is worth practicing writing clear and concise answers to questions like this about the key topics in the module, without referring to the lecture notes.]

Sort the array [5, 3, 4, 6, 8, 4, 1, 9, 7, 1, 2] using Quicksort with the pivot chosen to be the middle (rounded down) element of the array at each stage, and a partitioning algorithm that leads to a stable sort. Say how your partitioning algorithm works, and show the state of the array at each stage, i.e. its order and partitioning for each recursive call.

Question 3 (20 marks)

Suppose you have an array of integers `a` with no duplicates. Write an efficient procedure `makeBalBST(a)` that creates a perfectly balanced binary search tree from them. You can use procedures `size(a)` that returns the size of an array `a`; `quicksort(a)` that returns the sorted version of array `a`; `makeBT(v,l,r)` that returns a binary tree with root value `v`, left binary sub-tree `l` and right binary sub-tree `r`; and `aPart(a,i1,i2)` that returns an array made up of elements `i1` to `i2` of array `a`. [Hint: Think carefully about how many times, if any, you should call `quicksort(a)`to make your algorithm most efficient. You will probably find it easiest to use recursion, but not necessarily using repeated calls of your main `makeBalBST(a)` procedure.]

What is the overall average time complexity of your algorithm?

Question 4 (20 marks)

Outline, in no more than 100 words, the general Mergesort procedure for sorting an array.

Sort the array [4, 7, 8, 2, 3, 1, 2, 3, 6, 5] using Mergesort. Show the state of the array at each stage, i.e. its order and partitioning for each recursive call.

Question 5 (22 marks)

Explain, in no more than 150 words, when and why applying two phase Radix Sort to an array is able to produce a sorted array. State any conditions that must be satisfied for it to work well.

A library has its books organized primarily according to 20 categories represented by the two digit codes 01, 02, 03, … 20, and secondarily according to the first two letters of the first author’s surname Aa, Ab, …, Az, Ba, …, Zz. Use Radix Sort to sort the set of books with keys: [07 Ce, 09 Fa, 17 Mo, 09 Ce, 10 Fa, 09 Mo, 07 Aa, 07 Fa]. Show the state of the book list and what is being done at each stage.