CS2011 Tutorial Sheet 3: Algorithms and their Complexity

This is to occupy one tutorial session. You should write out answers to the exercises **before the tutorial**.

If you are having any difficulties with any of the material of the course, such as that on algorithms, their correctness and complexity, or on the languages C and SML, this may be a good time to raise them with your tutor.

Complexity

Be prepared to explain to your tutor the "big O" notation for the order of magnitude of a function and its role in expressing complexity measures of an algorithm.

Also, be prepared to explain how recurrence relations arise in calculating the complexity of recursive algorithms.

Heapsort

We look at the problem of sorting a list of items which support an order operation (eg integers with the usual less-than-or-equal-to) into ascending (non-descending) order.

There are algorithms for sorting which have the best possible time complexity: $O(N \log N)$ comparisons in the worst case ('mergesort' is such an algorithm). Other algorithms are space efficient: the so-called 'in-place algorithms' sort the list using only the list itself as storage ('bubblesort'is such an algorithm but has w.c. time complexity $O(N^2)$).

In this tutorial we introduce an efficient in-place algorithm whose worst-case time complexity of $O(N \log N)$. It is based on a data structure called a *heap*, which is a certain form of binary tree, and hence the algorithm is called *heapsort*.

Task 1.

Heaps are described in the course textbook *Fundamentals of Algorithmics* by Brassard and Bratley (ISBN 0-13-335068-1) 1996 (Section 5.7). Read this material and make sure you understand it.

Alternative descriptions may be found in (*Data structures and algorithm analysis in C*, M.A. Weiss) in Sections 6.1, 6.2 and 6.3 and Heapsort is in Section 7.5; and in *Algorithms* by R. Sedgewick (Addison-Wesley, Second Edition).

PTO

Task 2.

Describe each of the steps involved in making the following sequence into a heap. You may do so by drawing pictures of the trees at each step.

$$[2, 5, 4, 6, 4, 2, 1, 7, 9, 6]$$

The production of a heap from a sequence can be implemented in linear time $O(N)$ (as explained in the textbook).

Once you have produced a heap from this sequence, show the steps in constructing an ordered sequence by the heapsort algorithm. Again, you may do this by displaying the trees/arrays produced at each step. Be prepared to explain to your tutor why this is an $O(N \log N)$ operation.