CS341 Advanced Topics in Algorithms: Assignment 1

Due Thursday 9 February 2017 at 12noon

The maximum mark for this assignment is 20 points, which represents 10% of the total module credit. All answers need to be fully justified. The assignment is on two pages.

1. [4 points] Answer the following questions, giving a proof or a counterexample.

   (a) Given a sorting network diagram, we take its mirror reflection around a horizontal axis. Will this always result in a sorting network diagram?

   (b) Given a sorting network diagram, we take its mirror reflection around a vertical axis, and then flip the directions of all comparators, so that the minimum of every comparison is still returned on the left and the maximum on the right. Will this always result in a sorting network diagram?

   (c) Given a sorting network diagram, we turn it upside-down, and then flip the directions of all comparators so that the minimum of every comparison is still returned on the left and the maximum on the right. Will this always result in a sorting network diagram?

   (d) Given a sorting network diagram, we introduce a new comparator at an arbitrary position within the network. Will this always result in a sorting network diagram?

2. [4 points] Let a be an array of coloured balls of size n. Each ball is an object with a "colour" attribute, and possibly other data attributes. Design an efficient BSP algorithm to reorder the elements of a so that the balls appear in order of their colour. No particular order of balls within each colour is required. Solve the problem, if there are

   (a) only two colours, *red* $<$ *blue* (i.e. all the red balls must come before all the blue balls in the output array);

   (b) three colours, *red* $<$ *white* $<$ *blue* (the *Dutch national flag problem*).

   Give the algorithms' asymptotic BSP costs and state all necessary assumptions.

3. [4 points] The sequence of *Fibonacci numbers* $1, 1, 2, 3, 5, 8, \ldots$ is defined by a recurrence

$$f_0 = f_1 = 1 \qquad f_k = f_{k-1} + f_{k-2} \qquad k \geq 2$$

   Another recurrence for Fibonacci numbers can be given in matrix form:

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} f_{k-2} \\ f_{k-1} \end{pmatrix} = \begin{pmatrix} f_{k-1} \\ f_k \end{pmatrix}$$

   Design an efficient BSP algorithm that, given a value n, outputs an array of size n containing the first n Fibonacci numbers. Give the algorithms' asymptotic BSP costs and state all necessary assumptions.

4. [4 points] Consider the problem of sorting an array of objects by a certain attribute (the sorting *key*). A sorting algorithm is called *stable,* if it preserves the original order between any objects that have equal keys. The *radix sorting algorithm* sorts an array of m-digit integers by performing stable sorting first by the least significant digit, then by the second least significant digit, etc., and finally by the most significant digit. Justify the correctness of the radix sorting algorithm, and explain why its digit sorting subroutine needs to be stable. Design an efficient BSP algorithm to sort an array of size n of m-bit integers, based on radix sorting. Give the algorithm's asymptotic BSP costs and state all necessary assumptions.

5. [4 points] In some programming languages, a comment is introduced by the character '`%`': everything from that character up to (but not including) the end of the current line is ignored by the compiler. Note that the character '`%`' may appear many times within a single comment. A program is given by a character array of size n; the end of every line is represented by a special end-of-line character. There are no assumptions on the length of a line: lines may be very short or very long. Design an efficient BSP algorithm to replace the contents of every comment in the program by blanks, while keeping the rest of the program intact. Give the algorithm's asymptotic BSP costs and state all necessary assumptions.