CS341 Advanced Topics in Algorithms: Assignment 2

Due Thursday 9 March 2017 at 12noon

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

1. [2 points] Let $a = (a_0, \ldots, a_n)$ be a sequence of $n + 1$ positive integers. Consider computing the product of n rectangular matrices $A_0 \cdot A_1 \cdot \ldots \cdot A_{n-1}$, where for every i, A_i is an $a_i \times a_{i+1}$ matrix (i.e. it consists of a_i rows and a_{i+1} columns). The cost of computing a product of an $k \times l$ and an $l \times m$ matrix is taken to be klm for all k, l, m. Show that for $n \geq 3$, the cost of evaluating the product of n matrices may depend on the order of evaluation (e.g. the costs of evaluating $A_1A_2A_3$ as $(A_1A_2)A_3$ or $A_1(A_2A_3)$ may be different). Given an array of matrix dimensions a as input, the *matrix chain product* problem asks for minimum possible cost of evaluating the corresponding matrix product. Design an efficient BSP algorithm solving the matrix chain product problem. Give the algorithm's asymptotic BSP costs and state all necessary assumptions.

2. [2 points] Let a be a number array of size n. A subset of k indices $0 \leq i_0 < i_1 < \ldots < i_{k-1} < n$ is said to define an *alternating subsequence*, if $a_{i_0} < a_{i_1}$, $a_{i_1} > a_{i_2}$, $a_{i_2} < a_{i_3}$, $a_{i_3} > a_{i_4}$, ... Design an efficient BSP algorithm to determine the length k of the longest alternating subsequence of array a. Give the algorithm's asymptotic BSP costs and state all necessary assumptions.

3. [2 points] A group of n children sit in row. Every child is given a numerical rating of their school performance. The ratings are all different and are represented by a number array of size n. The teacher wants to give at least one sweet to every child; if two children sit next to one another, the one with a higher rating must get more sweets than the other. Design an efficient BSP algorithm to determine the number of sweets given to every child, so that the total number of sweets is minimised. Give the algorithm's asymptotic BSP costs and state all necessary assumptions.

4. [2 points] String r is said to be a *supersequence* of string s, if s is a (not necessarily contiguous) subsequence of r. A string is said to be a *common supersequence* of strings s, t, if it is a supersequence of both s and t. Let strings a, b be represented by character arrays of size n. Design an efficient BSP algorithm to determine the length of the shortest common supersequence of a and b. Give the algorithm's asymptotic BSP costs and state all necessary assumptions.

5. [2 points] The *Game of Life* is concerned with the evolution of a pattern of cells on a regular unordered 2D grid. In each *generation*, a cell can be either *dead* or *alive*. The cell's state in the next generation depends on the current states of the cell's eight neighbours (vertical, horizontal and diagonal). A dead cell that has exactly three alive neighbours becomes alive in the next generation. An alive cell that has two or three alive neighbours remains alive in the next generation. Otherwise, the cell dies. Design

an efficient BSP algorithm for running the Game of Life in parallel, on an $n \times n$ grid for n generations. Give the algorithm's asymptotic BSP costs and state all necessary assumptions.