THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2015
Campus: City

COMPUTER SCIENCE

Algorithm Design and Analysis

(Time allowed: Fifty minutes)

NOTE: Attempt *all* questions. Put the answers in the space below the questions. Write clearly! You may continue your answers onto the "overflow" pages provided at the end of the test, if necessary.
Use of calculators is NOT permitted.
This test is worth 10% of your final grade for the course.

SURNAME:

FORENAME(S):

STUDENT ID:

Student Name: ____________________ Student ID: ____________________

1. Consider an instance of the Stable Matching Problem in which the preferences of the blue nodes B_1, B_2, B_3 are respectively $P_1 > P_2 > P_3, P_2 > P_3 > P_1, P_3 > P_1 > P_2$, while the preferences of the pink nodes P_1, P_2, P_3 are respectively $B_3 > B_2 > B_1, B_2 > B_1 > B_3, B_1 > B_3 > B_2$.

   (i) Show the execution of the Gale-Shapley algorithm (with blue nodes as proposers). List every engagement made and every engagement broken, and the final output. [5 marks]

   B_1 engages P_1; B_2 engages P_2; B_3 proposes P_1; P_1 breaks engagement; P_1 engages B_3; B_1 engages P_3. Final assignment has $(B_1, P_3), (B_2, P_2), (B_3, P_1)$.

   (ii) Does there exist a solution to the problem that has each B_i matched with P_i? Explain why or why not. [3 marks]

   Yes. In each pair at least one gets its top choice so has no incentive to change, hence this is a stable assignment.

   (iii) Explain why in any solution, B_2 must be matched with P_2. [2 marks]

   Each is the other's top choice so if paired in any other way they would form a blocking pair.

Student Name: ______________________ Student ID: ______________________

2. Suppose that I have a new algorithm for multiplication of matrices that works by dividing each of the $n \times n$ matrices x and y into 3 parts of as equal size as possible, and computing the product xy by means of 23 multiplications of the parts, plus 147 matrix additions and subtractions.

   (i) Write down a recurrence describing the worst-case running time of this algorithm on an instance of size n. [5 marks]

   $T(n) \leq 23T(n/3) + g(n)$ where g takes the overhead into account. Note that the size of the problem we take to be n, so the 9 submatrices each have size about $n/3$. If $\leq$ is replaced by $=$, it is probably OK (no marks lost). Here $n/3$ makes sense only when n is a multiple of 3. Since g is increasing, we could write $T(\lceil n/3 \rceil)$ instead. I am not worrying much about these minor points, since they turn out not to affect the asymptotic solution. Nor is the initial condition very important.

   (ii) Is this algorithm likely to be faster than the standard matrix multiplication algorithm when used on large inputs? Give full explanation. [5 marks]

   Yes. According to the "master theorem" covered in class, the divide and conquer recurrence has solution in $\Theta(n^{\log_3 23})$. This is because the overhead is of order n^2. In order to compute the matrix product in the standard way, we require $\Theta(n)$ operations to compute each entry of the product, and hence $\Theta(n^3) = \Theta(n^{\log_3 27})$ overall and this is asymptotically worse than the running time of the new algorithm.

Student Name: ______________________________ Student ID: ______________________________

3. Recall that an *Egyptian fraction* is a rational number of the form $1/n$, where n is a positive integer. Consider the problem of writing a positive rational number less than 1 as a sum of *different* Egyptian fractions. For example, $2/3 = 1/2 + 1/6$ is a valid representation, but $2/3 = 1/3 + 1/3$ is not valid.

   (i) State a greedy algorithm for this problem. Show that it always finds a solution to the problem. You must show why the algorithm terminates. [5 marks]

   If x is the first denominator of the Egyptian fraction chosen for m/n, then we have $x - 1 < n/m \leq x$. Thus after one iteration we have $m/n - 1/x = (mx - n)/(nx)$ and the numerator is less than m by above. So the numerators form a strictly decreasing sequence of natural numbers, which terminates. But this algorithm can terminate if and only if we reach 0.

   (ii) Carry out the algorithm on the rational number $14/15$. [3 marks]

   We get $14/15 = 1/2 + 1/3 + 1/10$ by repeatedly choosing the largest Egyptian fraction we can.

   (iii) Show that the greedy algorithm does not always find the optimal solution, if we seek to minimize the number of summands. Hint: consider numbers of the form $(a + 1)/na$. [2 marks]

   $8/77 = 7/77 + 1/77 = 1/11 + 1/77$ from hint; however the greedy algorithm gives $8/77 = 1/10 + 3/770$ and hence will yield a longer expansion since $3/770$ is not in reduced form.

Student Name: ______________________ Student ID: ______________________

OVERFLOW PAGE

Student Name: ______________________________ Student ID: ______________________________

OVERFLOW PAGE