UNSW
>014676109

The University of New South Wales
Session 1 Examination 2013

COMP3121/3821
Algorithms and Programming Techniques

and

COMP9101/9801
Design and Analysis of Algorithms

- Time allowed: **3 hours**
- 3121/9101 students: Answer questions 1-4
- 3821/9801 students: Answer questions 1-6
- Each problem is worth as indicated in the parentheses
- You **can** keep the examination paper
- Write **LEGIBLY**
- You **MUST JUSTIFY** each answer
- **No** study material or textbooks are allowed
- UNSW approved calculators are **permitted**

1	2	3	4	5 (extended)	6(extended)

TOTAL:

ANSWERS MUST BE WRITTEN IN INK. EXCEPT WHERE THEY ARE EXPRESSLY REQUIRED, PENCILS MAY BE USED ONLY FOR DRAWING, SKETCHING OR GRAPHICAL WORK.

COMP3121/3821/9101/9801 Final Exam (session 1, 2013)

1. You are given an $n \times n$ chessboard with an integer in each of its n^2 squares. You start from the top left corner of the board; at each move you can go either to the square immediately below or to the square immediately to the right of the square you are at the moment; you can never move diagonally. The goal is to reach the right bottom corner so that the sum of integers at all squares visited is minimal.
   a) Describe a greedy algorithm which attempts to find such a minimal sum path and show by an example that such a greedy algorithm might fail to find such a minimal sum path. (5pt)
   b) Describe an algorithm which always correctly finds a minimal sum path and runs in time n^2. (10pt)
   c) Describe an algorithm which computes the number of such minimal paths.(5pt)

2. A palindrome is a sequence of at least two letters which reads equally from left to right and from right to left.
   a) Given a sequence of letters S, find efficiently its longest subsequence (not necessarily contiguous) which is a palindrome. Thus, we are looking for a longest palindrome which can be obtained by crossing out some of the letters of the initial sequence without permuting the remaining letters.(15pt)
   b) Find the total number of occurrences of all subsequences of S which are palindromes of any length ≥ 2. (be careful when counting!!)(25pt)

3. You are given n boxes of cookies $B_1, \ldots, B_n$. All cookies are of the same size; each box contains cookies of one kind only, but different boxes contain cookies of different kinds. The capacity of a box B_i is c_i cookies, ($c_i \geq 2$), and each box is filled to its capacity. You have to repackage cookies into the same boxes so that each box contains at most two cookies of the same kind, no box contains more cookies than its capacity and the number of leftover cookies (if any) is minimal. (20pt)

4. Assume that you are the administrator of a network of computers; each computer is connected by unidirectional fiber-optic cables of the same capacity to a few other computers on the same network (so the network can be modeled by a directed graph). You noticed that computers $P_1, P_2, \ldots P_n$ are mounting an attack on a computer P_0; since it is a real emergency, you must disconnect some of the optical cables of the network so that none of computers $P_1, P_2, \ldots P_n$ can send packets to P_0. Since you must send crews to disconnect some of the fibeoptic cables, for each cable c_{ij} for traffic from a computer P_i to a computer P_j there is an associated cost c_{ij} for disconnecting it. Your task is to design an algorithm for determining which cables to disconnect to isolate computer P_0 from all of the computers $P_1, \ldots, P_n$ so that the total cost incurred is minimal. (30pt)

Additional problems for the extended classes ONLY on the next page!

ADDITIONAL PROBLEMs FOR STUDENTS TAKING 3821 OR 9801 ONLY!

5. You are given a biased coin, but you do not know what the probability is to get a head when tossing it. Design an algorithm which uses this coin to simulate a dice, i.e., which produces outcomes $1 - 6$, all with equal probability. What is the expected number of of times you have to toss the unfair coin for each simulation of a throw of your "virtual fair dice"? (15pt)

6. Assume that you are given a fair coin. Is there a natural number K so that it is always possible to simulate a fair dice always using exactly K tosses of such a fair coin for each throw of your "virtual fair dice"? (15pt)