**Assessed Coursework** **Advanced Algorithms**

due: Thursday, 11 April, 2019 at 5:00pm **MA421**

## Instructions

The questions on the following pages describe assessed coursework, constituting 25% of the final mark for this course.

It is due on **Thursday, 11 April, 2019 at 5:00pm**.

One of the questions asks you to write a Java program. Please submit only a single final version of your program. Otherwise, you risk having the wrong submission marked.

You are required to submit solutions to the problems typeset with LaTeX (as a pdf file). A template LaTeX file is provided for the solutions.

The deadline is sharp. Late work carries an automatic penalty of 5 deducted marks (out of 100) for every 24 hours that the coursework is late.

Anonymous submissions will be handled by the course Moodle site. Use the corresponding link on the Moodle site to submit your solutions. Ideally, your submission should be comprised of two files: a single Java file that contains your whole program and a PDF file that contains your solutions to the problems. You will be able to submit *only once.*

Submission of answers to this coursework is **mandatory**. Without any submission, your mark for this course is incomplete, and you may not be eligible for the award of a degree.

The work should be yours only. No resource other than the course material is to be used. Please note the School's rules on Plagiarism and Assessment Offences as stated at here and at here.

The contents of your work must remain **anonymous**, so do not write your name on any submitted material. Instead, you must identify your work with your **candidate number** (5 digits, this is **not** your student number). You can check your candidate number via 'LSE for You'.

1. In this question, you are asked to implement Karger's Randomised Minimum Cut algorithm in JAVA. Your program should receive the input from the command line as

   `java RMinCut` $v_1\ v_2\ v_3\ v_4\ \cdots$

   where each argument is an integer and the arguments are to be processed in pairs to indicate the edges of the input graph: that is, the edges of the graph are $v_1v_2$, $v_3v_4$, and so on. If there is an odd number of arguments, the last one is to be ignored. There is no a priori bound on the number of vertices in the graph and the labels of the vertices are not necessarily a list of consecutive integers. Your algorithm should repeat Karger's basic algorithm sufficiently many times to achieve an error probability at most 0.01, and output the size of the minimum cut found and the set of vertices on each side of the cut.

   Your program should not use any special Java facility other than `Math.random()`, which returns a uniformly random value between 0 (inclusive) and 1 (exclusive), `Math.log(x)`, which computes the natural logarithm function, and `ArrayList` data structure.

2. The Bin Covering problem is defined as follows. For $n$ items with the given sizes $a_1, a_2, \ldots, a_n \in (0, 1]$, maximise the number of bins used so that each bin has items summing to at least 1.

   Give an asymptotic PTAS for this problem when restricted to the problem instances in which item sizes are bounded from below by some fixed constant $c > 0$.

3. A *guarded graph* $G$ is a tuple $(V, E, S, f)$, where $V$ is a set of nodes, $E$ is a set of edges over nodes in $V$ (that is, $E \subseteq \binom{V}{2}$), $S \subseteq V$, and $f : S \to \mathbb{N}$ is a function. You can view $S$ as the positions of security guards that are placed on a subset of nodes in the graph. Each guard reports the number of suspicious activities they observe in the neighbouring nodes, as represented by the function $f$. A *suspicious activity list* for a guarded graph $G = (V, E, S, f)$ is a subset $A$ of $V \setminus S$ such that, for each $v \in S$, $f(v) = |N(v) \cap A|$, where $N(v)$ denotes the neighbours of node $v$. The Guard Consistency problem is defined as

   $$\{\langle V, E, S, f\rangle \mid (V, E, S, f) \text{ is a guarded graph that has a suspicious activity list}\}.$$

   Prove that the Guard Consistency problem is $\mathcal{NP}$-complete.

4. Suppose you are given a biased coin that has $\Pr[\text{HEADS}] = p$, where $p$ is a constant unknown to you. The goal is to estimate $p$ with as few coin flips as you can. Assume you are given $b$, with $b \le p$, a lower bound for $p$. Devise a procedure for estimating $p$ by a value $\tilde{p}$ such that you can guarantee $\Pr[|p - \tilde{p}| > \delta p] \le \epsilon$, for any choice of the constants $0 < b, \epsilon, \delta < 1$.

5. Recall that a 3-CNF boolean formula is a conjunction of clauses, each of which is a disjunction of exactly 3 literals. In the Restricted-3-SAT problem, the goal is, given a 3-CNF formula $\phi$ with no negated variables (that is, when all the literals are positive), deciding whether there exists a truth assigment to the variables of $\phi$ such that there is exactly one true variable in each clause. Show whether the Restricted-3-SAT problem is polynomial-time solvable or $\mathcal{NP}$-complete.