Algorithms CMSC-37000 Final Exam. March 18, 2014

Show all your work. **Do not use text, notes, or scrap paper.** When describing an algorithm in pseudocode, **explain the meaning of your variables** (in English). WARNING: the bonus problems are underrated. This exam contributes 45% to your course grade. **Take this problem sheet home** for your amusement.

1. (2 points) **(Spell)** Spell the singular of "vertices." Print your answer.

2. (12 points) Given a graph, find a maximal (not necessarily maximum) matching in linear time. Describe your algorithm in pseudocode. (A *matching* is a set of pairwise disjoint edges. It is *maximal* if no edge can be added to it.)

3. (15 points) Let K be an n-bit integer. We are given a function $g : \{1, 2, \ldots, K\} \to \{0, 1\}$ by a black box: we can feed an integer x ($1 \leq x \leq K$) to the black box and it produces the value $g(x)$.

   Suppose $g(1) = 0$ and $g(K) = 1$. Find a value y such that $g(y) = 0$ and $g(y + 1) = 1 \quad (1 \leq y \leq K - 1)$. Use as few queries to the black box as possible. State the number of queries made in terms of n. Describe your algorithm in elegant pseudocode.

4. (8+25+6 points) **(Spanning tree)** **(a)** Define the min-cost spanning tree problem (input, output). Make sure you specify the conditions the input needs to satisfy. Does the problem involve directed or undirected graphs? Does the input need to specify a root? **(b)** Jarník's (a.k.a. Prim's) algorithm grows a tree from a start node. Describe the algorithm in pseudocode. **(c)** Name the three abstract data structure operations required for the implementation of the algorithm.

5. (28+40 points) **(Knapsack)** Recall that the Knapsack problem takes as input $2n + 1$ positive real numbers $w_1, \ldots, w_n, v_1, \ldots, v_n$, and W (weights, values, and weight limit) and asks to maximize the quantity $\sum_{i \in I} v_i$ over all choices of the set $I \subseteq [n]$ satisfying the constraint $\sum_{i \in I} w_i \leq W$.

   (a) Let $V = \sum_{i=1}^{n} v_i$. Assuming all *values* v_i are integers, find the optimum value in $O(nV)$ steps. Define what one "step" means in this

statement. Define your variables (the "brain" of your algorithm). Describe your algorithm in pseudocode. Name the method used.

(b) Prove that for $\epsilon > 0$ one can solve the Knapsack problem with real input variables (both the weights and the values are real) within a factor of $(1-\epsilon)$ of the optimum (i.e., to find a set I that satifies the constraint exactly and produces a value at least $(1-\epsilon)$-times the optimum) in $O(n^3/\epsilon)$ steps. Define what one "step" means in this statement. Describe your algorithm in clearly organized English. Prove that the desired approximation factor is achieved.

6. (2+4+20 points) **(Sorting networks)** **(a)** Define the concept of sorting networks.
(b) Prove that a sorting network requires $\gtrsim 2\log_2 n$ parallel steps to sort n items.
(c) State and prove the $(0,1)$-principle for sorting networks.

7. (32 points) **(Sandwich)** Find three languages L_1, L_2, L_3 over the same alphabet such that $L_1 \subset L_2 \subset L_3$ and $L_2 \in \mathrm{P}$ while L_1 and L_3 are undecidable.

8. (28 points; lose 9 points for each mistake) Consider the following three statements: (A) 3-colorability of graphs can be decided in polynomial time. (B) RSA can be broken in polynomial time. (C) Integers can be factored into their prime factors in polynomial time. – Which of the six implications is known: (A) $\Rightarrow$ (B); (B) $\Rightarrow$ (A); (A) $\Rightarrow$ (C); (C) $\Rightarrow$ (A); (B) $\Rightarrow$ (C); (C) $\Rightarrow$ (B). Do not prove. Make sure your answers are clearly readable. Put your answers in two columns: a "YES" column and a "NO" column. Each implication must appear in exactly one column.

9. (15 + 24 points) **(Finding the median)** In class we found the median of n numbers using $O(n)$ comparisons. The algorithm started with dividing the n items into groups of 5. (a) Would the same method work with groups of 7? (b) Would it work with groups of 3? State the recurrence obtained for each case and state and prove the resulting estimate of the number of comparisons. Ignore rounding.

10. (40 points) **(Vertex cover)** A *3-hypergraph* $H = (V, E)$ consists of a set V of vertices and a set E of edges; each edge is a set of 3 vertices. A subset $C \subseteq V$ is a *vertex cover* if it intersects every edge. H is given by a list of edges. Assume we are also given a non-negative weight for each vertex: $w : V \to \mathbb{R}$. The **weighted vertex cover** problem asks to find a minimum-weight vertex cover. Find a factor-3 approximation of the optimum using linear programming (i.e., find a vertex-cover of which the weight is at most 3 times the optimum).

11. (24+7+3 points) **(Wrong NP)** Larry defines a class of languages we call LNP as follows: The language $L \subseteq \Sigma^*$ belongs to LNP if
$(\exists$ finite alphabet $\Sigma_1)(\exists L_1 \subseteq \Sigma_1^*,\ L_1 \in P)(\exists c \in \mathbb{N})(\forall x \in \Sigma^*)(\exists w \in \Sigma_1^*)(x \in L \leftrightarrow (|w| \leq |x|^c$ and $(x, w) \in L_1))$.
(a) Prove that LNP is the set of all languages. (b) Make the minimum change to Larry's definition to turn it into the definition of NP. (c) Expand the acronym "NP" (state the English expression indicated by these two letters).

12. (20+4+12+6 points) **(Modular exponentiation)** Given the positive integers a, b, m, compute the quantity $a^b \pmod m$ in polynomial time. **(a)** Describe your algorithm in ELEGANT pseudocode. Your algorithm must NOT make recursive calls and must NOT make explicit use of the binary expansion of b. **(b)** Name the method used. **(c)** State the loop invariant from which the correctness of the algorithm immediately follows. **(d)** If Alice wants to send Bob an RSA-encrypted message and Bob wishes to decrypt it, who needs to perform modular exponentiation?

13. (5+20+8B+8B points) **(Boolean functions)**

    (a) What is the number of Boolean functions in n Boolean variables?

    (b) Construct a 3-CNF formula which is NOT satisfiable. (Each clause involves 3 distinct variables.) Make your formula as short as possible. Prove that your formula is indeed not satifiable.

    (c) (BONUS) Prove: almost all 3-CNF formulas with $m = 7n$ clauses are not satifiable. Here n is the number of variables. A random clause is obtained by selecting a triple of distinct variables at random and assigning each variable either itself or its negation by flipping three coins. A random 3-CNF is the AND of m independently chosen random clauses (so repetition is possible). (Checknote: the size of the sample space is $(8\binom{n}{3})^m$.)

    (d) (BONUS) Find an explicit Boolean function in $n \geq 4$ variables which cannot be represented as a 3-CNF formula. Your function must have a very simple (mathematical) description. Prove.

14. (15+15 points) **(Large numbers)** **(a)** Given $n \geq 1$, prove that $n!$ cannot be computed in polynomial time. Clearly state the two relevant quantities about which you claim that one is not polynomially bounded as a function of the other. **(b)** Can the quantity $n^{\lfloor \log n \rfloor}$ be computed in polynomial time? Prove your answer.

15. (18+8B points) **(Good assignments)** Given a 3-CNF formula with m clauses, a "good assignment" is an assignment of Boolean values to the variables that satisfies at least $7m/8$ of the clauses. (Each clause

involves 3 distinct variables.) **(a)** Prove: a good assignment always exists. If you use random variables, state the size of the sample space you are referring to. Explain your notation. **(b)** (BONUS) Prove: the probability that a random assignment satisfies at least $7m/8$ clauses is at least $1/(m+1)$.

16. (BONUS: 10B points) Describe an $O(n)$-time algorithm that, given a set S of n distinct real numbers and a positive integer $k \leq n$, determines the k numbers in S closest to the median of S.