6.045 Final Exam Solutions

Your Name: _______________________

May 22, 2012

Last Minute Tips:

• **DON’T PANIC.** Tackle everything on the test. Don’t skip questions because they look “scary”: once you figure out what they’re asking, you’ll probably realize they’re a lot easier than they seemed!

• **Don’t be a novelist.** Every part of the free-answer questions can be answered in a few sentences or less. *If you’re writing paragraph after paragraph, you’re wasting your time!* Your goal is just to convince us that you “get it,” not to impress us with your writing skills or attention to detail.

• **Budget your time.** Don’t spend more than about 15 minutes on one problem unless you’ve already finished the others.

• **Go over your responses in the true/false/open section.** You’d be amazed at the number of careless mistakes people make.

*Good luck, and may the power of Turing be with you!*

<table>
<thead>
<tr>
<th>True/False/Open</th>
<th>/ 20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem 1</td>
<td>/ 10</td>
</tr>
<tr>
<td>Problem 2</td>
<td>/ 10</td>
</tr>
<tr>
<td>Problem 3</td>
<td>/ 10</td>
</tr>
<tr>
<td>Problem 4</td>
<td>/ 10</td>
</tr>
<tr>
<td>Problem 5</td>
<td>/ 15</td>
</tr>
<tr>
<td>Problem 6</td>
<td>/ 10</td>
</tr>
<tr>
<td>Problem 7</td>
<td>/ 15</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>/ 100</td>
</tr>
</tbody>
</table>
(Proven) True, (Proven) False, or Open—Circle One [20 points, 1 point per statement].

(a) T F O : NP is contained in BQP.
(b) T F O : If P ≠ NP then RSA is secure.
(c) T F O : If RSA is secure then P ≠ NP.
(d) T F O : Quantum computers can break all public-key cryptosystems.
(e) T F O : For all f, g, h, if f = O(g(n)) and g(n) = Ω(h(n)) then f(n) = Θ(h(n)).
(f) T F O : No consistent formal system can contain a proof of its own inconsistency.
(g) T F O : Every NP-hard problem belongs to NP.
(h) T F O : If pseudorandom generators exist, then so do secure public-key cryptosystems.
(i) T F O : If P = NP, then P^A = NP^A for every oracle A.
(j) T F O : The gate set {NOT, MAJ} is universal, where MAJ(x, y, z) = 1 iff x + y + z ≥ 2.
(k) T F O : There are recognizable languages that are neither decidable nor Turing-equivalent to HALT.
(l) T F O : There are NP languages that are neither in P nor NP-complete.
(m) T F O : K(x) (Kolmogorov complexity) is Turing-reducible to BB(n) (the Busy Beaver function).
(n) T F O : There exists a constant c such that |K(x) + K(y) − K(x#y)| ≤ c for all strings x, y (where # means concatenation).
(o) T F O : Every language decidable by a PDA is also decidable by an NDFA.
(p) T F O : If ZPP = BQP then RP = coRP.
(q) T F O : There are at most countably many languages not reducible to the halting problem.
(r) T F O : ODDFAC = \{N | N has an odd number of prime factors\} is NP-complete.
(s) T F O : The language \{⟨M⟩ | there exists an x such that M(x) halts\} is Turing-equivalent to SUPERHALT.
(t) T F O : If A and B are both context-free, then so is A ∩ B.
Problem 1: Regular and Context-Free Languages [10 points].

(a) [5 points] Give a regular expression for the language

\[ L = \{ x \in \{0, 1, a, b\}^* \mid \text{all 0's are to the left of all 1's, and all a's are to the left of all b's} \} \].

\[(0|a)^* ((0|b)^*(1|a)^*) (1|b)^*\]

(b) [5 points] Give a context-free grammar for the language \( L = \{a^n b^m c^{n+m} \mid n, m \geq 0\} \).

\[
\begin{align*}
S & \rightarrow aSc \mid T \\
T & \rightarrow bTc \mid \varepsilon
\end{align*}
\]
Problem 2: Country Clubs [10 points]

(a) [5 points] Call a program $P$ a self-arbiter if it accepts $\langle P \rangle$ (i.e., its own code) and rejects all other strings other than $\langle P \rangle$. Does there exist a self-arbiter? If so, give pseudocode for one; if not, give a proof of impossibility.

Here is a self-arbiter:

Accept if the input equals the following string repeated twice, second time in quotes:
“Accept if the input equals the following string repeated twice, second time in quotes:"

Let $L(P)$ be the set of strings accepted by program $P$ (note that $P$ need not halt on all inputs). Also, call a subset $S$ of $\{0, 1\}^*$ a country club if for all programs $P$,

$\langle P \rangle \in S \quad \text{iff} \quad L(P) = S.$

Intuitively, a country club consists of exactly those programs able to recognize membership in the country club.

(b) [5 points] Is the empty set a country club? Why or why not?

It is not, because there exist programs $P$ such that $L(P) = \emptyset$ (for example the program that just rejects everything), but these programs are not elements of $\emptyset$.

(c) [Extra credit, 5 points] Does there exist a country club $S$? If so, describe one; if not, give a proof of nonexistence.

There doesn’t exist a country club. Suppose for the sake of contradiction that a country club $S$ did exist. We saw in part (b) that $S \neq \emptyset$, so there exists $\langle P \rangle \in S$. By definition, $L(P) = S$, in other words, $P$ recognizes the language $\{ \langle P \rangle | L(P) = S \}$. But this is not a recognizable language, as the following many-one reduction from HALT proves:

Given $\langle (M), x \rangle$, construct a TM $M'$ that works as follows: on input $y$, $M'$ runs $M$ on $x$ in parallel with $P$ on $y$ (alternating steps). If $P$ rejects $y$, $M'$ continues running $M$ on $x$. If $M$ halts on $x$, or if $P$ accepts $y$, $M'$ accepts.
Problem 4: Nonlocal Games [10 points].

A “nonlocal game” $G$ is a game played by two cooperating players, Alice and Bob. The game is defined by:

- A finite set $Q$ of questions,
- A finite set $A$ of possible answers,
- An “evaluation function” $V : Q \times Q \times A \times A \rightarrow \{0, 1\}$.

The game works as follows: first a referee chooses arbitrary questions $x, y \in Q$, and sends $x$ to Alice and $y$ to Bob. Then Alice sends back a response $a = a(x)$ depending only on $x$, and Bob sends back a response $b = b(y)$ depending only on $y$. Alice and Bob “win” the game if $V(x, y, a, b) = 1$.

Call the game $G$ “winnable” if there exist functions $a : Q \rightarrow A$ and $b : Q \rightarrow A$ such that $V(x, y, a(x), b(y)) = 1$ for all $x, y \in Q$.

Let NLGAME be the language consisting of all (binary encodings of) winnable nonlocal games. (Here, you can take the “size” $n$ of a game to be $|Q|^2 |A|^2$, the number of bits needed to specify $V$’s truth table.)

Show that NLGAME is NP-complete. [Hint: One way to do it is to reduce from 3-COLORING.]

Clearly NLGAME $\in$ NP, since a yes-witness is just the truth tables of $a : Q \rightarrow A$ and $b : Q \rightarrow A$ (of $O(|Q| \log |A|)$ bits).

To show that NLGAME is NP-hard, we give a reduction from 3-COLORING: given a graph $G = (V, E)$ we produce a nonlocal game $H$ with

- $Q = V$ (i.e., the questions are the vertices of $G$),
- $A = \{R, G, B\}$ (i.e., the answers are the three colors),
- $V(x, y, a, b)$ is defined as follows:
  - $V(x, y, a, b) = 0$ if $x = y$ and $a \neq b$ (“consistency check”)
  - $V(x, y, a, b) = 0$ if $(x, y) \in E$ and $a = b$ (“edge check”)
  - $V(x, y, a, b) = 1$ otherwise

We can construct $G$ in time $O(|V| + |E|)$, so the reduction is polynomial. If $G$ is 3-colorable, then a winning strategy is simply for Alice and Bob to both answer a common, valid 3-coloring for $G$. On the other hand, suppose $G$ is not 3-colorable. Because of the consistency check, we must have $a = b$ (i.e. $a(x) = b(x) \forall x \in G$) if Alice and Bob win with probability 1. But this means that there must exist an $(x, y) \in E$ for which the edge check fails—i.e. $a(x) = b(x)$. Hence $H \in$ NLGAME iff $G \in$ 3-COLORING.
Problem 5: SAT-Solver That’s Fast if P=NP [10 points].

Suppose that someone managed to prove \( P = NP \), but the proof was nonconstructive: that is, it didn’t say how to actually find polynomial-time algorithms for \( NP \)-complete problems like 3SAT. Show that just knowing that \( P = NP \) is good enough! More precisely, describe an explicit, deterministic algorithm \( A \) with the following property:

- If \( P = NP \), then there exists a polynomial \( p \) such that \( A(\varphi) \) outputs a satisfying assignment for any satisfiable 3SAT instance \( \varphi \), after at most \( p(|\varphi|) \) steps.

Briefly explain why your algorithm works. [Hint: Use dovetailing.]

We saw in one of the psets that search and decision are equivalent: if there is a polynomial-time algorithm for 3SAT, then there is also a polynomial-time algorithm for finding a satisfying assignment to a given 3SAT formula. We’ll find it by dovetailing.

For \( i = 1, 2, 3, \ldots \):

- For each \( j = 1, \ldots, i \): generate the \( j \)-th algorithm \( A_j \) in lexicographic order (it’s just a binary string), and run it on \( \varphi \) for \( j \) steps.
- If the algorithm outputs something, check if it satisfies \( \varphi \), and if so output it. Otherwise move on to the next algorithm.

Since we’re assuming that \( P = NP \), there’s some index \( j \) such that \( A_j \) is a polynomial-time algorithm for 3SAT. Let \( q \) be the polynomial representing the running time of \( A_j \). Our dovetailing algorithm will run \( A_j \) on \( \varphi \) for \( q(|\varphi|) \) steps in iteration \( i = \max \{ j, q(|\varphi|) \} \), and will then find the satisfying assignment. All the work we do before then is \( O \left( \sum_{k=1}^{i} k^2 \right) = O(q^3(|\varphi|)) \), so the total runtime is still polynomial. (Note that \( j \) is a constant, so it does not affect the asymptotic runtime.)
Problem 6: Collision Problem [25 points].

The “collision problem” arises frequently in cryptography. In the most common version, we are given black-box access to a function $f : \{1, \ldots, M\} \to \{1, \ldots, M\}$, where $M$ is even and typically extremely large (i.e., exponential). By “black-box access,” we mean that the only way to learn about $f$ is to call a subroutine that returns $f(x)$ for a given input $x$.

We are promised that $f$ is either one-to-one (that is, a permutation) or two-to-one (that is, for all $x$, there is exactly one $y \neq x$ such that $f(y) = f(x)$). The problem is to decide which holds. There are no other promises about $f$.

(a) [5 points] How many evaluations of $f$ are needed by a deterministic algorithm to solve the collision problem, in the worst case over $f$? Give an exact answer in terms of $M$. Briefly explain why your answer is both an upper and a lower bound.

(b) [5 points] How many evaluations of $f$ are needed by a randomized algorithm to solve the collision problem, with probability at least (say) $2/3$ over the algorithm’s internal randomness, again in the worst case over $f$? Give an asymptotic answer in terms in $M$.

Briefly explain why your answer is both an upper and a lower bound.

(c) [10 points] Describe a zero-knowledge interactive protocol, by which a prover Merlin (who knows $f$) can convince a skeptical verifier Arthur that $f$ is one-to-one rather than two-to-one, with high probabilistic confidence.

In your protocol, Arthur should need to make only $O(1)$ queries to $f$.

Briefly explain why your protocol satisfies the conditions of completeness, soundness, and zero-knowledge.

(d) [5 points] Let $N$ be a product of two large prime numbers, $p$ and $q$. Let $a$ be a uniformly random integer from 2 to $N−1$ that’s relatively prime to $N$. For all integers $x$, define $f(x) = a^x \pmod{N}$. Suppose you had a fast algorithm for finding collision pairs, meaning $x \neq y$ such that $f(x) = f(y)$.

Explain, informally, how such an ability could be helpful for factoring $N$, and thereby breaking the RSA cryptosystem.

(e) [Extra credit, 5 points] Consider the following quantum algorithm for the collision problem. First prepare a superposition

$$\frac{|1\rangle + \ldots + |M\rangle}{\sqrt{M}}.$$ 

Next, query $f$ in superposition, to obtain the state

$$\frac{|1\rangle|f(1)\rangle + \ldots + |M\rangle|f(M)\rangle}{\sqrt{M}}.$$ 

Finally, measure the $f(x)$ register.

What sort of state will be left in the $|x\rangle$ register if $f$ is one-to-one? What sort of state will be left if $f$ is two-to-one?

(f) [Extra credit, 5 points] Does your answer to (e) imply that a quantum algorithm can solve the collision problem using few queries to $f$? If so, why? If not, what else would be needed? How does the collision problem differ from Simon’s problem discussed in class?

(a) $M/2 + 1$ evaluations are necessary and sufficient: they’re sufficient because after $M/2 + 1$ queries we’ve either seen some value repeat (in which case $f$ is two-to-one) or we have seen $M/2 + 1$ unique values, which means that $f$ cannot be two-to-one. They’re necessary, because if an algorithm makes a set $Q$ of at most $M/2$ queries, it can’t distinguish a function that is one-to-one from a function that is actually two-to-one but pairs every element from $Q$ with an element that is not in $Q$. 


(b) $\Theta(\sqrt{M})$ evaluations are necessary and sufficient, because of the Birthday Paradox: each pair has, roughly, a $\frac{1}{M}$ probability of being a collision pair if $f$ is two-to-one, and there are $\left(\frac{\sqrt{M}}{2}\right) = \Theta(M)$ pairs. The lower bound follows from the union bound: let $p$ be the probability of finding a collision after $\leq T$ queries. Then $p \leq \sum_{1 \leq i < j \leq T} \Pr[f(x_i) = f(x_j)] \leq \left(\frac{T}{2}\right) \frac{1}{M}$ assuming $f$ was chosen uniformly at random from all two-to-one functions (and regardless of the values of $x_i$). So if we want $p \geq \frac{2}{3}$, then we need $T = \Omega(\sqrt{M})$.

(c) Arthur chooses a random element $x \in \{1, \ldots, M\}$, queries $f(x)$, and sends $f(x)$ to Merlin. Merlin sends back an element $y \in \{1, \ldots, M\}$, and Arthur accepts iff $x = y$.

Completeness: if $f$ is one-to-one (i.e., a permutation), Merlin can determine $x$ just from knowing $f(x)$, so he can convince Arthur with probability 1.

Soundness: if $f$ is two-to-one, there is another element $z$ such that $f(x) = f(z)$. Merlin can’t tell if Arthur chose $x$ or $z$, so he fails with probability at least $1/2$.

Zero-knowledge: Arthur is able to simulate the entire interaction by himself, because if $f$ is one-to-one, Merlin just sends him back $x$, the value Arthur himself selected.

(d) Suppose we find $x \neq y$ such that $a^x \equiv a^y \pmod{N}$. Then $a^{x-y} \equiv 1 \pmod{N}$. By Euler’s Theorem, this means that $x = y$ must divide $\phi(N) = N - p - q + 1$. Now, we know from a pset that if we can find $\phi(N)$ then we can also find $p$ and $q$. By taking the least common multiple of several independent $(x, y)$ pairs, we can get $\phi(N)$ with high probability.

(e) If $f$ is one-to-one, we’ll get a classical basis state $|x\rangle$ in the first register after we measure $|f(x)\rangle$. If $f$ is two-to-one, we’ll get an equal superposition $\frac{|x\rangle + |y\rangle}{\sqrt{2}}$ where $f(x) = f(y)$.

(f) No, this doesn’t yet solve the collision problem, since to learn that $f$ is two-to-one, we would need to measure the superposition $\frac{|x\rangle + |y\rangle}{\sqrt{2}}$ in a way that somehow revealed info about both $x$ and $y$. The collision problem differs from Simon’s problem by being less structured: we don’t have the promise that $f(x) = f(y) \Rightarrow x = y \oplus s$ for a single “secret string” $s$. 
Problem 7: GHZ Game [15 points].

The GHZ (Greenberger-Horne-Zeilinger) game is an interesting variant of the Bell/CHSH game from class and pset6. In the GHZ game, there are three players, Alice, Bob, and Carol, who can agree on a strategy in advance but are out of communication with each other once the game starts. Alice, Bob, and Carol receive bits \(x, y,\) and \(z\) respectively, where the triple \((x, y, z)\) is chosen uniformly at random from among the following four possibilities:

\[(0, 0, 0), (0, 1, 1), (1, 0, 1), (1, 1, 0).\]

The players’ goal is to output bits \(a, b,\) and \(c\) respectively such that

\[a + b + c \pmod{2} = x \text{ OR } y \text{ OR } z.\]

(a) [5 points] Describe a deterministic, classical strategy by which Alice, Bob, and Carol can win the GHZ game with probability 3/4.

(b) [Extra credit, 5 points] Show that no classical strategy does better than your strategy from (a).

(c) [5 points] Let

\[|\psi\rangle = \frac{|000\rangle - |011\rangle - |101\rangle - |110\rangle}{2}.\]

Calculate the result if Hadamard gates are applied to the first and second qubits of \(|\psi\rangle\) (but not the third qubit).

(d) [5 points] Suppose that at the start of the game, each player holds one qubit of \(|\psi\rangle\). Suppose each player adopts the following rule: “If my input is 0, then I’ll measure my qubit in the \{\(|0\rangle, |1\rangle\}\} basis and output whatever I see. If my input is 1, then I’ll first apply a Hadamard gate to my qubit, then measure my qubit in the \{\(|0\rangle, |1\rangle\}\} basis and output whatever I see.”

Using this strategy, with what probability will the players win the game? Justify your answer. [Hint: You can appeal to symmetry between players, rather than saying the same thing three times.]

(a) Alice and Bob output 0 and Carol outputs 1, ignoring \(x, y,\) and \(z.\) This will cause them to win in all cases except \(x = y = z = 0,\) i.e. with probability \(\frac{3}{4}.\)

(b) By convexity, we can consider deterministic strategies only. (A probabilistic strategy will just be a mixture of deterministic ones, so it can’t do better.) We need \(a(x) \oplus b(x) \oplus c(x) = x \lor y \lor z.\) This means that for all \(x, y, z\) where \(x \oplus y \oplus z = 0:\)

\[
\begin{align*}
a(1) \oplus b(1) \oplus c(0) &= 1 \\
a(1) \oplus b(0) \oplus c(1) &= 1 \\
a(0) \oplus b(1) \oplus c(1) &= 1
\end{align*}
\]

Adding these last three equations mod 2 yields

\[a(0) \oplus b(0) \oplus c(0) = 1\]

which contradicts the first equation. Therefore, any deterministic strategy fails in at least one of the four cases.
(c) Applying a Hadamard gate for the first qubit yields:

$$|0 - 0⟩ - |0 + 1⟩ + |1 + 0⟩ + |1 - 1⟩$$

(omitting normalization for convenience, and using the $|+⟩$ and $|−⟩$ shorthand for the $\frac{|0⟩ ± |1⟩}{\sqrt{2}}$ qubits). Then, applying a Hadamard gate to the second qubit yields:

$$|010⟩ - |001⟩ + |100⟩ + |111⟩$$

(d) If $x = y = z = 0$, then Alice, Bob, and Carol all measure in the standard basis, so they output a random $a,b,c$ such that $a \oplus b \oplus c = 0$.

If $x = y = 1$ but $z = 0$, then by part (c), Alice, Bob, and Carol will output a random $a,b,c$ such that $a \oplus b \oplus c = 1$.

By symmetry between Alice, Bob, and Carol, the same holds in the cases $x = z = 1$ but $y = 0$ and $y = z = 1$ but $x = 0$.

Therefore, in all four cases, the players win the GHZ game with probability 1.