COMS 461 - Midterm 3 Review

1. (8 points) Is the following Boolean formula satisfiable? Explain.

$$(x \vee y) \wedge (x \vee \bar{y}) \wedge (\bar{x} \vee y) \wedge (\bar{x} \vee \bar{y})$$

2. (16 points) A Boolean formula in conjugate normal form can be converted to a multivariable polynomial with integer coefficients so that the Boolean formula is satisfiable if and only if the polynomial has an integer root. Here is how:

   - Convert any non-negated variable x to $(1-x)^2$.
   - Convert a negated variable $\bar{x}$ to x^2.
   - Convert each $\vee$ into multiplication $(*)$.
   - Convert each $\wedge$ into addition $(+)$.

   (a) Use these steps to convert the Boolean formula

   $$(x \vee y) \wedge (x \vee \bar{y}) \wedge (\bar{x} \vee y) \wedge (\bar{x} \vee \bar{y})$$

   into an integer polynomial. You do not need to simplify the polynomial.

   (b) Let INTEGER-ROOT $= \{\langle p \rangle : p$ is a multivariate integer polynomial with an integer root$\}$. Given that we can convert Boolean formulas to integer polynomials as described above, which of the following is true?

   A. SAT $\leq_p$ INTEGER-ROOT.
   B. INTEGER-ROOT $\leq_p$ SAT.
   C. Both. INTEGER-ROOT and SAT are polynomial-time equivalent.
   D. Neither. SAT $\not\leq_p$ INTEGER-ROOT and INTEGER-ROOT $\not\leq_p$ SAT.

   (c) It turns out that INTEGER-ROOT is Turing recognizable, but not decidable. Which of the following complexity classes does INTEGER-ROOT belong to? If it is in more than one, choose the smallest one.

   A. P
   B. NP
   C. NP-complete
   D. EXP
   E. NP-hard

3. (16 points) For each of the following, determine if the statement is true or false. Briefly explain your answers.

   (a) $\log n \in O(n)$.

   (b) $n^2 \log n \in O(n^2)$.

   (c) $n^3 + \log n \in O(n^3)$.

   (d) $n^n \in O(2^{(n^2)})$.

4. (12 points) Let $f : \{1, \ldots, N\} \to \{1, \ldots, N\}$ be an invertible function such that when these integers are represented in binary, f can be computed in polynomial time, but f^{-1} cannot be computed in polynomial time. Let

$$L = \{(x, y) \,:\, f^{-1}(x) \leq y\}.$$

   (a) Explain why $L \in$ NP.

   (b) The assumptions above imply that $L \notin P$. Why is this not a proof that P $\neq$ NP?

      A. Because one example is not enough to prove that P $\neq$ NP.
      B. Because L might not be NP-complete.
      C. Because there might not be a function f with the properties described above.
      D. Because L is not decidable.

5. (8 points) What does the Cook-Levin theorem say about SAT?

6. (12 points) Let NO-REPEATS $= \{\langle A \rangle : A$ is an integer array with no repeat entries$\}$.

   (a) The following algorithm decides if an array A of integers is in NO-REPEATS. What is the Big-O run time of this algorithm? Use n to denote the length of the input (encoded in binary). You can assume that any two integers with binary lengths less than n can be compared in $O(n)$ time.

```
let k = length(A)
# Loop through pairs of entries in A to see if any are the same:
for i from 1 to k-1:
  for j from i+1 to k:
    if A[i] == A[j]:
      return False
# If you don't find any entries of A that are the same:
return True
```

   (b) Which of the following complexity classes does NO-REPEATS belong to? If it is in more than one, choose the smallest one.

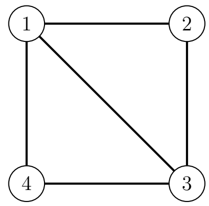
      A. P
      B. NP
      C. NP-complete
      D. EXP
      E. NP-hard

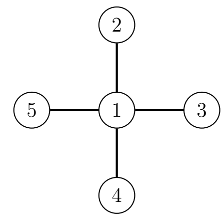
7. (8 points) A graph is 3-colorable if you can assign one of three colors to each of the vertices in such a way that no edge connects two vertices with the same color. Let

$$\text{3-COLORABLE} = \{\langle G \rangle : G \text{ is a 3-colorable graph}\}.$$

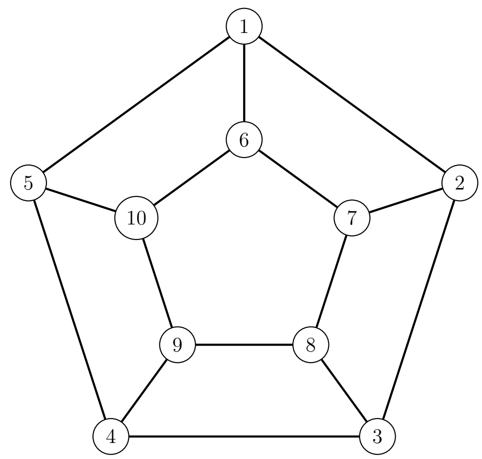
Is it possible to create a nondeterministic polynomial-time algorithm to decide 3-COLORABLE? You do not have to find an algorithm, just explain why such an algorithm does or does not exist.

8. (20 points) A graph G has a Hamiltonian cycle if there is a path that (i) starts and ends at the same vertex, and (ii) visits every other vertex exactly one. For example the graph on the left below has a Hamilton cycle ($1 \to 2 \to 3 \to 4 \to 1$), but the graph on the right does not.





(a) Find a Hamiltonian cycle for the graph below (there is more than one right answer).



(b) Let HAMILTON-CYCLE $= \{\langle G \rangle : G$ is a graph with a Hamilton cycle$\}$. Prove that the language HAMILTON-CYCLE $\in$ NP by describing a polynomial-time verifier.

(c) The language HAMILTON-CYCLE is actually NP-complete. What does this mean about its relationship with the language SAT?

A. SAT $\leq_p$ HAMILTON-CYCLE, but HAMILTON-CYCLE $\not\leq_p$ SAT.

B. HAMILTON-CYCLE $\leq_p$ SAT, but SAT $\not\leq_p$ HAMILTON-CYCLE.

C. SAT $\leq_p$ HAMILTON-CYCLE and HAMILTON-CYCLE $\leq_p$ SAT.

D. HAMILTON-CYCLE $\not\leq_p$ SAT and SAT $\not\leq_p$ HAMILTON-CYCLE.