CSC 364 **Quiz #2** **November 8/2000**

Name ________________________________ Student No. ________________________________

NO AIDS ALLOWED

Answer ALL questions on test paper. Use backs of sheets for scratch work.

Total Marks: 35

(1) (5 points) Define the complexity class NP. Be as specific as possible.

NP is a collection of languages. A language L is in NP if there exists a polynomial-time Turing machine, R, with the following properties. First, R takes two inputs (x, y), and R runs in polynomial time in $|x| + |y|$. Secondly, there exists a constant c such that (i) for all $x \in L$, there exists a y, $|y| \leq |x|^c$ such that $R(x, y)$ accepts; and (ii) for all $x \notin L$, for all y, $R(x, y)$ rejects.

(2) (10 points) Design a Sipser-basic Turing machine (one head, and one tape infinite to the right) that takes a string of 0's and 1's as input and produces the reverse of the string as output. For example, if the tape is initially 11001 (followed by blanks), then after halting the tape should be 10011 (followed by blanks). Give the high-level description of the Turing Machine, and specify the states that you will use. You do not need to give a full formal description of the Turing Machine.

The high level control of M is as follows.

(a) First, M replaces its input w by $\$w\&$. The head should then be pointing at the & sign.

(b) Loop: Scan left to the first (meaning right-most) nonblank symbol. If the symbol is the $ sign, then go to step (c). Otherwise, remember its value, replace it with a blank, and scan right to the & sign, and then scan right some more until we hit the first blank symbol. Write the remembered value on the cell and scan left, back to the & sign.

(c) Overwrite with a blank, until we hit the & sign. When we hit the & sign, overwrite with a blank, move right and halt.

The states are as follows. Below, x and y always refer to the values 0 or 1. You do not need to give this much detail to get full credit.

(a) We start in q_0. The transitions are as follows.

$(q_0, x) \to ([a_1, x], \$, R)$.
$([a_1, x], y) \to ([a_1, y], x, R)$,
$([a_1, x], blank) \to ([a_2], x, R)$.
$([a_2], blank) \to ([b_1], \&, L)$,

(b) $([b_1], blank) \to ([b_1, blank, L)$
$([b_1], x) \to ([b_2, x], blank, R)$, where x is 0 or 1.
$([b_1], \$) \to ([c_1], blank, R)$
$([b_2, x], blank) \to ([b_2, x], blank, R)$
$([b_2, x], \&) \to ([b_3, x], \&, R)$
$([b_3, x], y) \to ([b_3, x], y, R)$
$([b_3, x], blank) \to ([b_4], x, L)$.
$([b_4], x) \to ([b_4], x, L)$
$([b_4, \&) \to ([b_4], \&, L)$.

(c) $([c_1], x) \to ([c_1], blank, R)$
$([c_1], blank) \to ([c_1, blank, R)$
$([c_1], \&) \to (q_accept, blank, R)$.

(3) The complement of a language $L \subseteq \{0,1\}^*$ consists of all strings $x \in \{0,1\}^*$ that are not in L.

a. (5 points) Suppose that an arbitrary langauge L is in P. Explain why the complement of L, $\overline{L}$ is also in P.

Suppose that M is a polynomial-time Turing machine accepting L. Then we know that for every string $x \in L$, M on x halts in state q_{accept}, and for every string $y \notin L$, M on y halts in state q_{reject}. Thus, we can create a polynomial-time Turing machine M' for the complement of L by simply making the old accept state the new reject state and making the old reject state the new accept state.

b. (5 points) If L is in NP, then does it also follow that the complement of L is in NP? Justify your answer briefly.

It does not follow that the complement of a language is also in NP. For example, 3SAT is in NP. The complement of 3SAT consists of those strings that are not encoded properly, plus those strings that encode a 3SAT instance that is not satisfiable. Note that an instance f of 3SAT that is satisfiable has a short witness, namely, a truth assignment where f evaluates to true. But on the other hand, a formula f that is NOT satisfiable has no obvious short witness.

(4) (10 points) IND-SET is the following problem. The input is an undirected graph $G = (V, E)$ and a number k. The input is accepted if and only if G contains an independent set of size k. An independent set of size k is a subset $V' \subseteq V$, $|V'| = k$, such that all edges within V' (that is, all edges (u', v'), such that $u', v' \in V'$) are NOT present in G. For example, the graph G1 below would be accepted for $k = 4$ and $G2$ would be rejected for $k = 4$.

a. (4 points) Show that IND-SET is in NP.

The Turing Machine $R(x, y)$ interprets x as the pair $(G = (V, E), k)$ and interprets y as a subset of V of size k. R accepts such an input if the strings x, y are properly encoded, and if the k nodes, V', specified by y form an independent set in G. R is clearly polynomial-time computable since one just has to check that all $\binom{k}{2}$ edges within V' are not present in G.

b. (6 points) Show that IND-SET is NP-complete by giving a polynomial-time reduction from the CLIQUE problem to IND-SET. Briefly justify your reduction. (Recall that an instance of CLIQUE is a graph $G = (V, E)$ and a number k. The instance is accepted if and only if G contains a clique of size k, where a clique of size k is a subset $V' \subset V$, $|V'| = k$ such that all edges within V' are present in G.)

We will give a polynomial-time reduction from CLIQUE to IND-SET. Let the input to CLIQUE be a pair $(G = (V, E), k)$. We form the graph $G' = (V, E')$ where an edge (i, j) is in G' if and only if it is NOT in G. Now we call IND-SET on the pair (G', k), and accept if and only if $IND - SET(G', k)$ accepts. This reduction is polynomial-time since each edge in the graph G' is obtained simply from G. It is left to show that G has a k-clique if and only if G' has a k-independent set. Suppose that G has a k-clique, on the vertices $V' \subseteq V$. Then all of the edges within V' will not be present in G' and hence V' will be an independent set in G'. By the same reasoning, if G' contains an independent set of size k on V', then G will contain a clique on V', thus completing our proof.