12 marks **Computational Complexity Theory** **1h20m**

Hardness is given as [x]. Higher number means harder. Marks as (x).

1. Describe why this machine that computes the successor of a natural number is not a TM: The machine takes as input a natural number n on a tape cell. The transition function is $\delta(q_0, n) = (q_1, n+1, \rightarrow)$ where q_1 is a halting state. [1] (1)
2. Define a TM that infinite loops on all inputs. [1] (1)
3. Is it possible for a TM to run for exactly $n-2$ steps and halt on all inputs of length n for all $n \geq 10$? If yes, describe such a TM. If no, prove none exists. [2] (1)
4. Prove that if a language is decided by a TM that has the ability to move left or right by two tape cells in one step in addition to the left or right movement, then the language is also decided by a TM without this ability. [2] (1)
5. A *turn* of the tape head of a TM is defined as a change in direction of the movement of the tape head. That is, a turn is when the tape head moves right and the previous move was a left or when it moves left and the previous move was a right. Suppose the tape head can only move left or right (no stay option). Describe an algorithm that takes as input (M, x, k) where M is a TM, x is an input to M, and k is a natural number. Your algorithm should output true if the number of turns of M on x is at most k and output false otherwise. Note that M can run in an infinite loop without adding any turns and in that case your algorithm should output true. [3] (1)
6. Show that P is closed under set union. [1] (1)
7. Show that NP is closed under set union. [1] (1)
8. Show that if NP $\neq$ coNP, then P $\neq$ NP. [1] (1)
9. For a language A, define $A{\downarrow} = \{x \in A \mid \text{for all } y \leq x \text{ s.t. } |y| \leq |x|,\ y \in A\}$. where $y \leq x$ is determined by interpreting binary strings x and y as binary representations of natural numbers. Show that $A{\downarrow} \in$ PSPACE if $A \in$ PSPACE. [2] (1)
10. Describe an algorithm that takes as input (M, x, k) where M is a TM, x is an input to M, and k is a natural number and outputs true if M on x uses at most k tape cells of space. Assume that M has a read-only input tape and a read-write tape and space is only counted when M writes (tape head is over) to some cell in the read-write tape. Two writes to the same cell only counts as one tape cell of space usage. Note that M can run in an infinite loop while consuming no additional space and in that case your algorithm should output true if the space consumed is less than k. [2] (1)
11. Show 4SAT $\leq_p^m$ 3SAT. Illustrate your reduction for
$$(x \vee y \vee \overline{z} \vee w) \wedge (\overline{x} \vee \overline{y} \vee z \vee \overline{w}).$$
[2] (1)
12. Suppose $A \leq_p B$ via an algorithm that queries B exactly on two strings and accepts if and only if both strings are accepted by B. Show that if $B \in$ NP, then $A \in$ NP. [3] (1)

Student's Scores
(To be filled by the instructor)

1	2	6	7	8	[1]
/1	/1	/1	/1	/1	/5

3	4	9	10	11	[2]
/1	/1	/1	/1	/1	/5

5	12	[3]
/1	/1	/2

Instructor's Remarks
(To be filled by the instructor)

Hints

1. The tape alphabet is a finite set. Natural numbers are not.
2. $\delta(q_0, .) = (q_0, ., R)$
3. No. Suppose a TM that runs for 8 steps for an input of length 10. Then, for any length 11 input that is an extension, the TM will run for 8 steps, not 9.
4. For $\delta(q,a) = (p, b, \leftarrow\leftarrow)$, we add $\delta(q,a) = (p', b, \leftarrow)$ and $\delta(p', .) = (p, ., \leftarrow)$.
5. Simulate $M(x)$ while keeping count of turns. We can reject when count is more than k. If some number of steps pass without any turn, then simulate until the current symbol is a blank. Let the current state be some q. Keep track of states until a turn is made. If some state is repeated before a turn, then $M(x)$ is in an infinite loop with no turns. Otherwise, the turn count is incremented and we are closer to deciding.
6. if $M_1(x)$ accepts, accept, else if $M_2(x)$ accepts, accept, else reject.
7. Take a string that encodes a pair (b, y) where $b \in \{1, 2\}$ and y is a string as input. The verifier for the union runs the verifier $M_b(x, y)$.
8. If P = NP, then coNP = P = NP as P is closed under complementation.
9. On input x, for each y with smaller length than x, check $y \leq x$ and $y \in A$ using PSPACE algorithm for A. If any $y \notin A$, reject, else accept. Note that there at at most $2^{O(|x|)}$ such y.
10. A TM that uses k tape cells can run for at most $2^{O(k)}$ steps. Compute an upper-bound on the number of steps from k and M. Then, run $M(x)$ while keeping track of space usage.
11. Replace $\ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4$ with $(\ell_1 \vee \ell_2 \vee a) \wedge (\ell_3 \vee \ell_4 \vee \overline{a})$ where we use a fresh variable a for each clause.
12. Suppose for input x, the strings queried on B are y and z. We can build a verifier for A as follows. We know $x \in A$ if and only if $y, z \in B$. Since $B \in$ NP, we know there are proofs y' and z' for their membership in B. So for proving $x \in A$, we can use (y', z').