Answer THREE questions, at least ONE from EACH of Sections A and B.

SECTION A

1. a. Explain how Turing machines can be encoded as strings over the alphabet $\{0, 1\}$.

[12 marks]

b. Explain what it means to say that a Turing machine *accepts* an input, and what it means to say that a Turing machine *accepts* a language. What does it mean to say that a language is *recursively enumerable*, and to say that a language is *recursive*?

[6 marks]

c. Let L be the language over $\{0, 1\}$ defined as follows,

$$\begin{aligned} L \quad = \quad & \{x \in \{0,1\}^\star : x \neq \text{code}(M) \text{ for any Turing machine } M \text{ or} \\ & x = \text{code}(M) \text{ and } M \text{ does not accept } x\} \end{aligned}$$

where for any Turing machine M, code(M) denotes the string over $\{0, 1\}$ obtained using the encoding you described in the first part of the question. Prove that L is not recursively enumerable.

[15 marks]

[Total 33 marks]

2. a. Define what it means to say that a language A *reduces* to a language B, and explain in detail why this concept is useful.

[10 marks]

b. Define *Post's correspondence problem* (PCP), and define the *modified Post's correspondence problem* (MPCP).

[5 marks]

c. Prove that MPCP reduces to PCP.

[18 marks]

[Total 33 marks]

3. a. Explain what it means to say that a Turing machine *accepts* a language. Why is it necessary to make a distinction between *accepting* and *recognizing* a language.

[5 marks]

b. Define the empty tape halting problem (ETHP). What does it mean to say that this problem is *unsolvable*.

[8 marks]

c. Prove, by reducing ETHP or otherwise, that the following decision problem is unsolvable.

Instance: Two Turing machines, M_1 and M_2.

Question: Let $L(M_1)$ and $L(M_2)$ be the languages accepted by M_1 and M_2. Is it the case that $L(M_1) \cap L(M_2) \neq \emptyset$?

[15 marks]

d. Give a brief outline of how a Turing machine designed to perform the reduction in your previous answer would operate. (Note: you do not have to design a machine, but you should give a basic description of the operations it needs to perform.)

[5 marks]

[Total 33 marks]

SECTION B

4. a. Describe how a *non-deterministic Turing machine* (NDTM) $M = (Q, \Sigma, q_0, \delta, F)$ works. Explain what it means when we say that a NDTM M solves the decision problem P. Make sure you include a clear definition of when M accepts an input and when it rejects its input.

[11 marks]

b. Define the class of non-deterministic, polynomial time problems (NP).

[5 marks]

c. Define the *3-satisfaction problem* for propositional logic (3SAT).

[5 marks]

d. Sketch an argument to show that 3SAT is in NP. If it helps, you may use a multi-tape machine for your argument.

[12 marks]

[Total: 33 marks]

5. Let A, B and C denote arbitrary decision problems.

   a. Define what it means to say that A *reduces to* B *in* p*-time* (in symbols, $A \leq B$).

   [7 marks]

   b. Let $\leq$ be the relation defined in part (a). Prove that $\leq$ is *transitive* i.e. if $A \leq B$ and $B \leq C$ then $A \leq C$.

   [6 marks]

   c. Let NP denote the class of non-deterministic, polynomial time decision problems. Using the ordering $\leq$ defined in part (a), or otherwise, define the class of all *NP hard problems*.

   [6 marks]

   d. Define the *Hamiltonian Circuit Problem* (HCP), *Travelling Salesman Problem* (TSP) and the *Travelling Salesman Decision Problem* (TSDP).

   [5 marks]

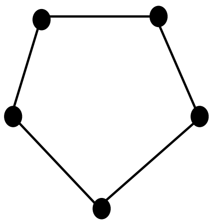
   e. Assume that the HCP is NP-hard (i.e. you do *not* have to prove this). Use that assumption and the transitivity of $\leq$ to prove that TSDP is also NP-hard.

   [9 marks]

[Total: 33 marks]

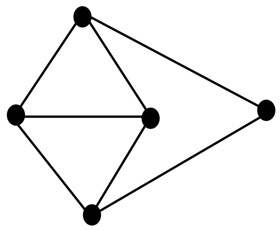
6. a. Consider the following *graph colouring problem* (GCOL). An instance of GCOL is any undirected, finite graph G together with a finite set of colours $C = \{c_1, c_2, \ldots, c_k\}$. (G, C) is a yes-instance if it is possible to colour each edge of G with some c_i ($1 \leq i \leq k$) such that no edge must be coloured by two distinct colours and such that two incident edges are never given the same colour. If G cannot be coloured in this way, (G, C) is a no-instance.

Which instances (below) are yes-instances of GCOL and which are no-instances? If you think it is a yes-instance, label a colouring of the graph.



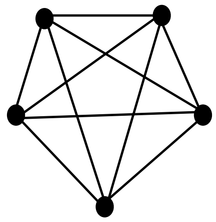
{red, green}

(a)



{red, yellow, blue}

(b)



{red, yellow, blue, green}

(c)

[5 marks]

b. Let (G, C) be any instance of GCOL. Write a formula $\phi(G, C)$ which is satisfiable if and only if (G, C) is a yes-instance of GCOL. [Hint: for each edge e of G and for $c \in C$ define a proposition $p_{e,c}$. The idea is that $p_{e,c}$ should be true if the edge e is coloured by c.] Deduce that GCOL reduces to PSAT in polynomial time.

[11 marks]

c. You are given that GCOL is NP hard. Use this to deduce that PSAT is also NP hard. You may assume that polynomial time reduction is transitive.

[8 marks]

d. Sketch an argument to show that GCOL is in NP.

[9 marks]

[Total: 33 marks]

END OF PAPER