Question 7.21 One can run a NDTM on a given input ϕ and count the number of its accepting branches to determine if $\phi \in$ *DOUBLE-SAT*. Therefore, *DOUBLE-SAT* is in NP. It remains to find a reduction from *3SAT* to *DOUBLE-SAT*. Let $\psi \in$ *3SAT* and construct $\psi^{'}$ as follows:

$$\psi^{'} = \psi \wedge (x \vee \bar{x}),$$

where x is a new variable not used in ψ. If ψ is satisfiable, $\psi^{'}$ has at least two satisfiable assignments.

Question 7.23

a One can construct a new boolean formula from any instance of CNF_2 in which a variable appears exactly once in positive and once in negative form. To do so, replace variables that appear only in positive form with *True* and remove the clauses that contain them. Similarly, for variables that appear only in negative form, replace them with *False* and remove their corresponding clauses. This does not affect the satisfiability of the given formula. Therefore, one can focus on boolean expressions of the following form:

$$\phi = (x_1 \vee \bar{x_2}) \wedge (x_2 \vee \bar{x_3}) \wedge ...(x_{n-1} \vee \bar{x_n})$$

The modified expression can be further reduced to perfect matching in a bipartite graph. To construct the graph, create a node for every variable x in ϕ. Furthermore, create a node for every clause c. Then for every clause $(x \vee y)$ add an edge from the clause node to variable nodes x and y. We provide the following lemma:
Lemma: The expression ϕ is satisfiable iff there is a perfect matching of size m in the corresponding bipartite graph G, where m is the number of clauses in ϕ.
Proof: Clearly, any satisfying assignment yields a matching of size m. To see why, assign a true literal, say x_i, to the clause c_k in which it appears positively. Similarly, assign a false literal, say x_j, to the clause c_l where it appears negatively. This constructs a matching since x_i and x_j appear once positively and once negatively in the clauses of ϕ.
Conversely, if there is a matching of size m in the graph, then a satisfying assignment can be constructed as follows: If there is an edge from the clause c_k to the node x_i, set $x_i = True$ if x_i appears positively in c_k; otherwise, set $x_i = False$. This assignment satisfies all clauses of ϕ.
Remark: Note that finding the perfect matching in a bipartite graph can be done in polynomial time (for instance, by computing the permanent of the adjacency matrix).

b Verifying that an assignment is satisfying and that the variables appear in at most 3 places can be done in polynomial time. Therefore, CNF_3 is in NP. Also, $3SAT$ can be reduced to CNF_3 by a change of variables. For every variable x that appears in $k > 3$ clauses, create new variables $x_1^{'}, x_2^{'}, ..., x_k^{'}$. Construct $\phi^{'} \in CNF_3$ as follows: replace x with $x^{'}$'s. Then, add new clauses that imply equivalence between $x_1^{'}, x_2^{'}, ...x_k^{'}$:

$$(x_1^{'} \vee \bar{x}_2^{'}) \wedge (x_2^{'} \vee \bar{x}_3^{'}) \wedge ... \wedge (x_{k-1}^{'} \vee \bar{x}_k^{'}) \wedge (x_k^{'} \vee \bar{x}_1^{'})$$

$\phi^{'}$ is in CNF_3 and is satisfiable iff ϕ is satisfiable.

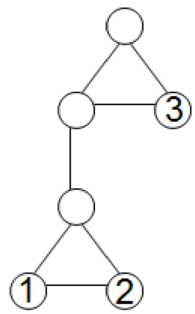

Figure 1: A clause gadget. Terminals 1,2, and 3 correspond to the first, second, and third literals respectively. If the top node is colored T, it means that the clause is satisfied.

Question 7.27 $3COLOR$ is in NP because a coloring can be verified in polynomial time. We show that $3SAT <_p 3COLOR$. Let $\phi = c_1 \wedge c_2 \wedge ... \wedge c_l$ be a 3cnf formula over the variables $x_1, ..., x_n$. We build a graph G with $2n+6l+3$ nodes, containing a variable gadget for each variable x_i, one clause gadgets for each clause, and one palette gadget as follows. Label the nodes of the palette gadget T, F, and R. Label the nodes in each variable gadget + and - and connect each to the R node in the palette gadget as shown in the hint. For each clause, create a gadget as shown in Fig.1.

Connect the top of the clause gadgets to the F and R nodes in the palette. Also, connect the top of its bottom triangle to the R node. For every clause c_j, connect the i-th ($1 \leq i \leq 3$) bottom node of its clause gadget to the literal node that appears in its i-th location. An example is shown below.

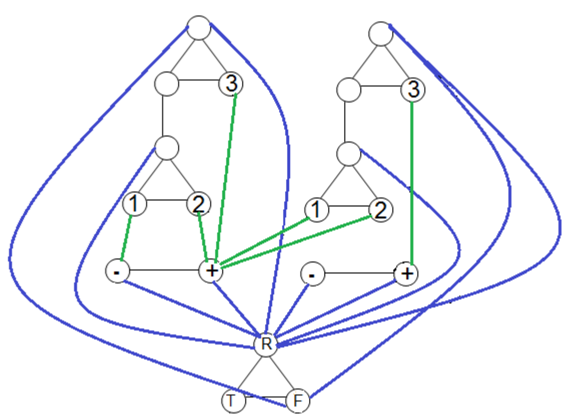

Figure 2: A graph constructed from $\phi = (\bar{x}_1 \vee x_2 \vee x_2) \wedge (x_1 \vee x_2 \vee x_2)$.

To show that the construction is correct, we first demonstrate that if ϕ is satisfiable, the graph is 3-colorable. The three colors are called T, F, and R. Color the palette with its labels. For each variable, color the + node T and the - node F if the variable is True in a satisfying assignment; otherwise reverse the colors. Because each clause has one True literal in the assignment, we can color the nodes of that clause so that the node connected

to the F node in the palette is not colored F. Hence we have a proper 3-coloring.
Similarly, if we are given a 3-coloring, we can obtain a satisfying assignment by taking the colors assigned to the + nodes of each variable. Observe that neither node of the variable gadget can be colored R, because all variable nodes are connected to the R node in the palette. Furthermore, if both bottom nodes of a clause gadget are colored F, the top node must be colored F, and hence, each clause must contain a true literal.

Question 7.28 *SET-SPLITTING* is in NP because we can verify in polynomial time that no subset C_i is monochromatic. To prove that the problem is NP-complete, we give a polynomial time reduction from 3SAT to SET-SPLITTING. Given an instance of 3SAT ϕ, set $S = \{x_1, \bar{x_1}, ..., x_n, \bar{x}_n, y\}$, where x_i's are the variables and y is a special color variable. The splitting is done as follows.
For every clause c_i in ϕ, let C_i be a subset of S containing the elements corresponding to the literals in c_i and the special element $y \in S$. Then $C = C_1, ..., C_k$.
If ϕ is satisfiable, consider a satisfying assignment. If we color all the true literals red, all the false ones blue, and y blue, then every subset C_i of S has at least one red element (because it is satisfiable) and it also contains one blue element y. This constitutes a splitting.
In addition, for a given splitting $< S, C >$, we can set the literals that are colored differently from y to true. Similarly, we set the literals that have the same color as y to false. This yields a satisfying assignment for ϕ.