Final Exam

1. (1 point) Is the following Boolean formula a tautology?

$$\big(((p \wedge \neg q) \to r) \wedge ((s \to \neg r) \vee (r \to \neg p))\big) \to ((\neg s \to q) \to q)$$

   If not, provide a falsifying assignment.

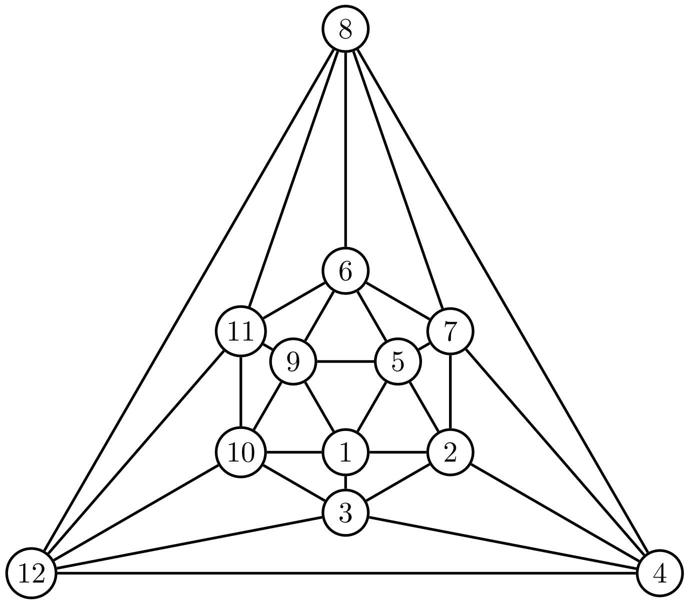
2. (1 point) Is the following first-order formula generally true?

$$\forall x \Big(\big(\forall y\, (R(x,y) \to (P(y) \vee Q(y)))\big) \to \big((\forall y\, (R(x,y) \to P(y))) \vee (\forall y\, (R(x,y) \to Q(y)))\big) \Big)$$

   If not, provide an interpretation of predicate symbols P, Q, and R on some set (domain) M, which makes the formula false.

3. (1 point) Does there exist an algorithm with polynomial delay that yields all correct colorings of a given graph G in 2 colors?

4. (2 points) Find the maximal k for which the graph below has an independent set of k vertices. Provide an example of such a set (as a list of vertex numbers) and prove that for larger values of k such an independent set does not exist.



5. (2 points) In this task, we consider undirected graphs without loops and parallel edges. A *graph homomorphism* from graph $G_1 = (V_1, E_1)$ to graph $G_2 = (V_2, E_2)$ is a mapping $f\colon V_1 \to V_2$ such that, for any pair of vertices u, v of G_1, if $(u,v) \in E_1$, then $(f(u), f(v)) \in E_2$. (In other words, a homomorphism is a mapping of vertices which preserves edges. This mapping is not required to be injective (i.e., two vertices of G_1 could map onto one vertex of G_2); neither it is required to be surjective (i.e., G_2 could include vertices not covered by this mapping).) Prove that the following problem is NP-complete: given two graphs G_1 and G_2, determine whether there is a homomorphism from G_1 to G_2.
   (*Hint:* try backwards reduction from a problem on graphs which you know to be NP-complete.)

6. (2 points) Suppose $\mathsf{P} \neq \mathsf{NP}$. Could there exist a polynomial-time algorithm which, being given a Boolean formula φ, answers whether it has *exactly one* satisfying assignment?
   (*Comment:* informal argumentation like "this problem is obviously harder than SAT, therefore not polynomially solvable" will not be accepted. There should be either a mathematical proof that the desired algorithm does not exist (e.g., by backwards reduction of a problem known to be hard), or a polynomial-time algorithm presented.)

(One extra point is added just for the fact of participating in the exam. Participating means submitting something within the scope of the deadline.)