Homework, the 4th series

Deadline: 29 May, 23:59.

Idea. Given a hypothetical algorithm deciding whether two labeled graphs are isomorphic, construct an algorithm, which actually *finds* such an isomorphism (if any), in a not much worse time.

Exact statement. A directed[1] *graph* of n vertices is given by a relation $E \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$. We encode it by a binary word w_E of length n^2, where the position $(i-1)n + j$ is 1 iff $E(i, j)$ holds.

A *labeled* graph is additionally equipped with a mapping $\ell : \{1, \ldots, n\} \to \mathbb{N}$. We encode it by

$$w_{E,\ell} \quad = \quad w_E \,\#\, \ell(1) \,\#\, \ell(2) \,\#\, \ldots \,\#\, \ell(n)$$

where the numbers $\ell(i)$ are given in binary.

Two labeled graphs with the same number of vertices n given by (E_1, ℓ_1) and (E_2, ℓ_2), respectively, are *isomorphic* if there exists a permutation $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$, such that

$$\ell_1(i) \quad = \quad \ell_2(\sigma(i)) \tag{1}$$

$$E_1(i, j) \iff E_2(\sigma(i), \sigma(j)). \tag{2}$$

We call such σ *isomorphism* and encode by $\sigma(1) \,\#\, \sigma(2) \,\#\, \ldots \,\#\, \sigma(n)$.

Assume that an algorithm A solves the decision problem, whether two labeled graphs are isomorphic in time $T(n)$. **Construct an algorithm** A', which in positive case finds an isomorphism σ. The algorithm A' should work in time $p(T(q(n)))$, for some polynomials[2] $p(n), q(n)$.

Bonus. An analogical question for non-labeled graphs. (Then isomorphism is just a permutation satisfying (2).)

Remark. For a completely correct solution of *Bonus*, you may obtain an additional 0.5 point (in the target scale), and thus double the score of this exercise.

[1] Consideration of *undirected* graphs does not affect the difficulty of the problem.

[2] In particular, if A works in polynomial time, so does A'.