XIII Летняя онлайн-школа «Комбинаторика и алгоритмы»
Даниил Мусатов, «Модели делегирования вычислений»
Список упражнений №1: классы **P**, **NP** и **coNP**

Через $\{0,1\}^*$ обозначается множество всех конечных слов из 0 и 1. *Языком* называется любое подмножество $\{0,1\}^*$. Язык называется *разрешимым*, если существует некоторый алгоритм $M$, такой что $M(x) = 1$ при $x \in L$ и $M(x) = 0$ при $x \notin L$.

Классом $\mathbf{TIME}(f(n))$ называется множество языков $A$, для которых существует некоторая константа $c$ и некоторый алгоритм $M$, такой что $M(x) = 1$ при $x \in L$, $M(x) = 0$ при $x \notin A$ и при всех $x$ вычисление $M(x)$ длится не более $cf(|x|)$ шагов. Классом **P** называется объединение $\bigcup_{k=1}^{\infty} \mathbf{TIME}(n^k)$, т.е. множество языков, разрешимых за полиномиальное время.

**1.** Докажите, что если $A$ и $B$ лежат в **P**, то $A \cup B$, $A \cap B$, $\overline{A}$ также лежат в **P**.

Классом **NP** называется множество языков $A$, для которых существуют полином $p(n)$ и алгоритм $V(x,y)$, такой что при всех $x \in A$ для некоторого $y$ верно $V(x,y) = 1$, при всех $x \notin L$ для всех $y$ верно $V(x,y) = 0$ и при всех $x$ и $y$ время работы $V(x,y)$ ограничено $p(|x|)$. (В частности, $V(x,y)$ прочтёт не больше $p(|x|)$ символов $y$).

**2.** Докажите, что если $A$ и $B$ лежат в **NP**, то $A \cup B$ и $A \cap B$ также лежат в **NP**.

**3.** Докажите, что $\mathbf{P} \subset \mathbf{NP}$.

Классом **coNP** называется множество $\{A \mid \overline{A} \in \mathbf{NP}\}$.

**4.** Докажите, что если $\mathbf{P} = \mathbf{NP}$, то $\mathbf{NP} = \mathbf{coNP}$.

**5.** На лекции мы обсудили, что задача проверки существования раскраски в 3 цвета лежит в **NP**. Но на практике может быть важно найти раскраску, а не просто проверить наличие. Пусть есть чёрный ящик, который по любому графу честно ответит, можно ли его покрасить в 3 цвета. Придумайте полиномиальный алгоритм, который позволит найти какую-нибудь раскраску.

На лекции мы обсудили, что с точки зрения делегирования вычислений наиболее прост класс $\mathbf{NP} \cap \mathbf{coNP}$: тогда, каким бы ни был ответ, его можно снабдить коротким и легко проверяемым доказательством.

**6.** Рассмотрим множество $\mathsf{PRIMES} = \{p \mid p \text{ простое}\}$. На лекции обсуждалось $\mathsf{PRIMES} \in \mathbf{NP}$, доказательство простоты строится так:

1. Предъявляется первообразный корень по модулю $p$, т.е. такой остаток $g$, что $g^{p-1} \equiv 1 (\mod p)$ и $g^n \not\equiv 1 (\mod p)$ для $n = 1, \ldots, p-1$.
2. Предъявляются разложение на простые множители $p - 1 = q_1^{\alpha_1} \cdot \cdots \cdot q_k^{\alpha_k}$, а также доказательства простоты всех $q_i$, построенные рекурсивно.
3. Верификатор проверяет, что $g^{\frac{p-1}{q_i}} \not\equiv 1 (\mod p)$ при всех $i$, а также все результаты рекурсивно.

Ответьте на следующие вопросы:

а) Почему достаточно проверить только $g^{\frac{p-1}{q_i}} \not\equiv 1 (\mod p)$, чтобы гарантировать $g^n \not\equiv 1 (\mod p)$ для $n = 1, \ldots, p-1$?

б) Оцените время работы всей рекурсии. Почему алгоритм будет полиномиальным (от $\log p$)?

**7.** Рассмотрим множество $\mathsf{FACTORING} = \{(n,a,b) \mid$ у числа $n$ есть простой делитель $p \in [a,b]\}$. Докажите, что $\mathsf{FACTORING} \in \mathbf{NP} \cap \mathbf{coNP}$. В этой задаче можно использовать без доказательства, что $\mathsf{PRIMES} \in \mathbf{NP} \cap \mathbf{coNP}$.