Лекция 13

Элементы теории сложности

(Конспект: О. Нескоромная)

13.1 Элементы теории сложности

13.1.1 Классы P и NP

Пусть Σ — конечный алфавит. Напомним, что массовая задача M есть некоторое множество индивидуальных задач — пар (u, s) (где $u, s \in \Sigma^*$, u — условие, s — решение).

Определение 13.1. $M \in \widetilde{\mathbf{NP}}$, если

1. M — *полиномиально ограничена*, т.е. существует многочлен p, такой, что для любого условия u, если существует хотя бы одно такое s, что $(u, s) \in M$, то существует и s' длины не более $p(|u|)$, такое что $(u, s') \in M$.

2. M — *полиномиально проверяема*, т.е. существует многочлен p, существует алгоритм A, такие, что $\forall u, s \in \Sigma^*((u, s) \in M \Leftrightarrow A(u, s) = 1)$ и при этом A заканчивает свою работу за время, не превосходящее $p(|u| + |s|)$.

Пример 13.1. $\{(N, m) \mid N \vdots m, 1 < m < N\}$.

Пример 13.2 ($\widetilde{\mathtt{SAT}}$ (*задача о выполнимости булевой формулы*)). Дана формула в конъюнктивной нормальной форме (конъюнкция конечного числа дизъюнкций, в каждую из дизъюнкций входят логические переменные либо их отрицания): например,

$$\{(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2) \wedge (x_4 \vee \neg x_2)\}.$$

Требуется найти значения переменных, такие, что значение всего выражения — истина (в приведенном примере: x_1 — истина, x_2 — ложь). Это задача из $\widetilde{\mathbf{NP}}$: решение — не длиннее условия, и подставить мы его также можем быстро. Формула, для которой такие значения существуют, называется *выполнимой*. □

Определение 13.2. Для каждой массовой задачи M определим язык $L(M) = \{u \mid \exists s\ (u,s) \in M\}$ — множество всех условий, для которых существуют решения.

Определение 13.3. Язык L принадлежит классу $\mathbf{NP}$, если $\exists M \in \widetilde{\mathbf{NP}}$: $L = L(M)$.

Определение 13.4. $M \in \widetilde{\mathbf{P}}$, если существует многочлен p и существует алгоритм A (который может выдавать строку из Σ^* или останавливаться с результатом «решения нет»), такие, что A работает не дольше, чем p(размер входа), и решает задачу M, т.е.

- $A(u) = s \implies (u,s) \in M$;
- $\exists s\ (u,s) \in M \implies A(u) \neq$ «решения нет».

Определение 13.5. $L \in \mathbf{P}$, если существует многочлен p и существует алгоритм A (выдающий 0 или 1), такие, что A работает не дольше, чем p(размер входа), и $\forall u \in \Sigma^*(A(u) = 1 \Leftrightarrow u \in L)$.

Открытый вопрос 13.1. $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$.

(Гипотеза: $\mathbf{P} \neq \mathbf{NP}$.)

13.1.2 Сводимости и полнота

«Упростим» этот вопрос, не изменяя его.

Определение 13.6. Язык L *полиномиально сводится* к языку L' (обозначим это $L \to L'$), если существует многочлен p и существует алгоритм A, работающий не дольше, чем p(длина входа), такие, что $\forall u \in \Sigma^*\ (A(u) \in L' \Leftrightarrow u \in L)$.

Определение 13.7. Язык называется $\mathbf{NP}$-*трудным*, если любой другой язык из $\mathbf{NP}$ к нему сводится. Язык называется $\mathbf{NP}$-полным, если он $\mathbf{NP}$-трудный и при этом сам принадлежит $\mathbf{NP}$.

Теорема 13.1. *Если L — $\mathbf{NP}$-полный и $L \in \mathbf{P}$, то $\mathbf{P} = \mathbf{NP}$.*

Доказательство. Очевидно. □

Теорема 13.2. `SAT` *(язык всех выполнимых формул) — $\mathbf{NP}$-полный.*

Следствие 13.1. *Если `SAT` $\in P$, то $\mathbf{P} = \mathbf{NP}$.*

13.1.3 Алгоритмы, использующие случайные числа

Определение 13.8. $M \in \widetilde{\mathbf{RP}}$, если

1. M — полиномиально ограничена.
2. M — полиномиально проверяема.
3. Каждое разрешимое условие M имеет не менее половины решений, т.е.

$$\forall u((\exists t(u,t) \in M) \Rightarrow$$
$$|\{s \mid (u,s) \in M,\ |s| \leqslant p(|u|)\}| \geqslant$$
$$\frac{1}{2} \cdot \text{кол-во всех строк длины не более } p(|u|))$$

(здесь $|\ldots|$ обозначает в одном случае — мощность множества, а в другом — длину строки; многочлен p — тот, что фигурирует в определении полиномиальной ограниченности).

Определение 13.9. $L \in \mathbf{RP} \Leftrightarrow \exists M \in \widetilde{\mathbf{RP}}\ L = L(M)$.

Очевидно, для задачи из $\widetilde{\mathbf{RP}}$ достаточно выбрать случайную строку длины $p(|u|)$, чтобы получить решение задачи u с вероятностью $\geqslant \frac{1}{2}$. Если повторить эту процедуру k раз, то вероятность успеха будет $1 - \frac{1}{2^k}$, чего для практических целей вполне достаточно.

Теорема 13.3. *Язык, состоящий из всех составных чисел, принадлежит* $\mathbf{RP}$.

Доказательство. Алгоритм, проверяющий простоту числа N:

- Если $N \vdots 2$ или $N = 1$, то сразу выдать правильный ответ.
- Случайно выбираем число M от 2 до $N - 1$.
- Если НОД$(M, N) \neq 1$, то выдать ответ «составное».
- (*) В противном случае, если $M^{(N-1)/2} \not\equiv (\frac{M}{N}) \pmod N$, то выдать ответ «составное».
- В противном случае, выдать ответ «возможно, простое».

(Здесь $(\frac{M}{N})$ — символ Лежандра.) Проверим корректность алгоритма. Все шаги корректны, проверим корректность шага (*) и то, что если число составное, то вероятность получить ответ не позднее этого шага — не менее $\frac{1}{2}$. Доказательство этого разобьем на следующие леммы (во всех из них предполагается, что N — нечетно и $\geqslant 3$).

Лемма 13.1. $N \in \mathbb{P} \Rightarrow M^{(N-1)/2} \equiv (\frac{M}{N}) \pmod N$.

Доказательство. Доказана в курсе алгебры. □

Лемма 13.2. *Если для всех M, взаимно простых с N, выполняется $M^{(N-1)/2} \equiv (\frac{M}{N}) \pmod N$, то $N \in \mathbb{P}$.*

Доказательство. Пусть $N \notin \mathbb{P}$, т.е. $N = p_1 \cdot \ldots \cdot p_k$. Рассмотрим 2 случая.

1. Среди p_i нет одинаковых. Пусть r — невычет по модулю p_1, т.е. $(\frac{r}{p_1}) \equiv -1 (\operatorname{mod} p_1)$. По китайской теореме об остатках существует M, такое, что $M \equiv r (\operatorname{mod} p_1)$, $M \equiv 1 (\operatorname{mod} p_i)$ при $i \neq 1$. Тогда $(\frac{M}{N}) = \prod_i (\frac{M}{p_i}) = -1$. С другой стороны, $M^{(N-1)/2} \equiv 1 (\operatorname{mod} N)$. Противоречие.

2. $N = p^2 n$ $(p \in \mathbb{P})$. Пусть r — первообразный корень по модулю p^2. Тогда $r^{N-1} = (r^{(N-1)/2})^2 \equiv (\frac{r}{N})^2 \equiv 1 (\operatorname{mod} r)$. Это значит, что $N-1 \vdots p(p-1)$. Но $N \vdots p$. Противоречие. □

Лемма 13.3. $N \notin \mathbb{P} \Rightarrow |\{M \in \{2, \ldots, N-1\} \mid M^{(N-1)/2} \not\equiv (\frac{M}{N}) \pmod N\} > \frac{N-2}{2}$.

Доказательство. Пусть сравнение выполняется для остатков $M_1, \ldots, M_k$ по модулю N. По лемме 13.2 существует M^*, для которого сравнение не выполняется.

Для остатков $M^* \cdot M_1, \ldots, M^* \cdot M_k$ по модулю N сравнение также не выполняется $((\frac{M_i M^*}{N}) = (\frac{M^*}{N}) \cdot (\frac{M_i}{N}) \not\equiv M_i^{(N-1)/2} (M^*)^{(N-1)/2} (\operatorname{mod} N))$. Кроме того, они все различны: $M_i M^* \equiv M_j M^* (\operatorname{mod} N) \Rightarrow M_i \equiv M_j (\operatorname{mod} N)$, поскольку НОД$(M^*, N) = 1$. Таким образом, их не меньше, чем тех, для которых сравнение выполняется. □

□