

Глава 1

Класс BPP .

1.1 Вероятностные вычисления за полиномиальное время.

Вероятностная машина Тьюринга – это машина Тьюринга с дополнительной лентой (только для чтения, головка движется только вправо), имитирующей (физический) датчик случайных чисел. Во всех ее клетках заранее записаны 0 или 1, причем предполагается, что это значения независимых испытаний случайной величины ξ с $Prob\{\xi = 0\} = Prob\{\xi = 1\} = 1/2$. Количество обращений к датчику в процессе вычисления есть величина зоны на этой ленте, в которой побывала головка; содержимое остальных клеток этой ленты не используется. Будем предполагать следующее:

- Вероятностная машина Тьюринга всегда останавливается и в качестве результата выдает 0 или 1.
- Время вычисления ограничено полиномом от длины входа. Случайная лента (в таком формализме) входной не является и учитывается при подсчете зоны вычисления. Как следствие, количество обращений к датчику случайных чисел также ограничено сверху полиномом от длины входа.
- Вероятностная машина Тьюринга M есть модель вычисления за полиномиальное время случайной величины $\zeta_x = M(x)$ с законом распределения

| | |
|--------------------|--------------------|
| 0 | 1 |
| $Prob\{M(x) = 0\}$ | $Prob\{M(x) = 1\}$ |

как функции от входного слова x . Специфической мерой сложности таких вычислений является количество обращений к датчику случайных битов $Q_M(x)$ – максимум числа использованных клеток дополнительной ленты-датчика, взятый по всем вариантам вычисления на входе x .

- Случайная величина ζ_x распознает язык $L \subset \Sigma^*$ с вероятностью ошибки $\alpha(n)$, если

$$\begin{aligned} x \in L &\Rightarrow \text{Prob}\{\zeta_x = 1\} > 1 - \alpha(|x|), \\ x \notin L &\Rightarrow \text{Prob}\{\zeta_x = 1\} \leq \alpha(|x|). \end{aligned}$$

Определение 1.1 Язык $L \subset \Sigma^*$ принадлежит классу BPP (от “bounded probabilistic polynomial”), если существует вычислимая за полиномиальное время случайная величина, распознающая L с постоянной вероятностью ошибки $\alpha(n) = 1/3$.

1.2 Частотные распознаватели.

Оказывается удобным переформулировать определение класса BPP в комбинаторных терминах без использования понятия случайной величины. Рассмотрим полином $q(n)$, ограничивающий сверху число обращений к датчику вероятностной машины M , распознающей язык L с вероятностью ошибки $\alpha(n)$.

Предикат $R(x, y)$

“ $|y| = q(|x|)$ и машина M на входе x с содержимым ленты-датчика y возвращает 1”

принадлежит классу P , причем

$$\text{Prob}\{M(x) = 1\} = \frac{|\{y \mid |y| = q(|x|), R(x, y) = 1\}|}{2^{q(|x|)}}.$$

Поэтому

$$\begin{aligned} x \in L &\Rightarrow \frac{|\{y \mid |y| = q(|x|), R(x, y) = 1\}|}{2^{q(|x|)}} > 1 - \alpha(|x|), \\ x \notin L &\Rightarrow \frac{|\{y \mid |y| = q(|x|), R(x, y) = 1\}|}{2^{q(|x|)}} \leq \alpha(|x|). \end{aligned} \tag{1.1}$$

Тройку (R, q, α) , где $R \in P$, q – полином, а α – любая функция, для которых выполнено (1.1), условимся называть *частотным распознавателем* языка L . Мы только что установили, что если язык распознается

на вероятностной машине Тьюринга с вероятностью ошибки α , то он обладает частотным распознавателем с той же α . Верно и обратное – частотный распознаватель легко переделать в вероятностную машину Тьюринга, которая будет распознавать тот же язык с той же вероятностью ошибки. Поэтому определение 1.1 эквивалентно следующему:

Определение 1.2 Язык $L \subset \Sigma^*$ принадлежит классу BPP , если он обладает частотным распознавателем (R, q, α) с $\alpha(n) = 1/3$.

Значение константы $1/3$ в этих определениях несущественно – класс не изменится, если вместо нее взять любое число c , $0 < c < 1/2$. За счет увеличения степени полинома q на 1 можно повысить надежность ответа гораздо больше:

Лемма 1.3 *Каждый язык L с частотным распознавателем $(R_0, n^d, 1/3)$ обладает также частотными распознавателями вида $(R_1, n^{d+1}, (2\sqrt{2}/3)^n)$, $(R_2, n^{d+2}, (2\sqrt{2}/3)^{n^2}), \dots$*

Доказательство. Вычисление значения предиката $R_1(x, z)$:

1. вычисляем $n := |x|$ и проверяем условие $|z| = n^{d+1}$ (если не равно, то результат 0);
2. разбиваем слово z на n кусков длины n^d каждый, $y_i := i$ -тый кусок;
3. $R_1(x, z) := \text{MAJORITY}(R_0(x, y_1), \dots, R_0(x, y_n))$, где MAJORITY – булева функция голосования (мнение большинства ее аргументов).

Частота правильного ответа для R_1 может быть вычислена, как вероятность более половины успехов в серии из n независимых испытаний в схеме Бернулли с вероятностью успеха $p = 2/3$. Она оценивается снизу величиной $1 - \lambda^n$, где $\lambda = 2\sqrt{p(1-p)} = 2\sqrt{2}/3$.

Конструкция R_2 аналогична, но слово z надо разбивать на n^2 кусков длины n^d каждый. ■

Замечание. Возможности вероятностного распознавания языков с вероятностью ошибки 0 оказываются теми же, что и у обычного детерминированного вычисления (т.е. датчик случайных битов ничему не

помогает). Это совершенно очевидно для формализма частотных распознавателей: для языка L , обладающего частотным распознавателем с нулевой вероятностью ошибки $(R, q, 0)$, справедливо

$$x \in L \Leftrightarrow \forall y_{|y|=q(|x|)} R(x, y) = 1 \Leftrightarrow R(x, 0^{q(|x|)}) = 1,$$

т.е. $L \in P$.

Замечание. То же самое имеет место и для более общего формализма вероятностных вычислений на машинах Тьюринга с дополнительной лентой-датчиком, отличающегося от приведенного выше отсутствием полиномиальных ограничений на время вычислений и число обращений к датчику случайных битов (требование к вычислению всегда заканчиваться – остается). В самом деле, если для некоторого x при некотором заполнении ленты-датчика бесконечной в обе стороны последовательностью y вероятностная машина M дает неверный ответ, то тот же (неверный) ответ она будет давать при заполнении ленты-датчика любой последовательностью z , совпадающей с y в битах с номерами $|i| \leq S_M(x)$. Мера множества таких последовательностей есть $2^{-2S_M(x)} > 0$. Поэтому вероятностное вычисление с нулевой вероятностью ошибки должно не давать ошибки ни при каком заполнении ленты-датчика. Это означает, что вместо случайного заполнения можно всю ленту-датчик заполнить (алгоритмически) нулями.

1.3 Включение $BPP \subset P/Poly$.

Мы видели, что каждый язык $L \in P$ обладает алгоритмом-распознавателем специального “схемного” вида:

Для выяснения принадлежности входного слова x языку L сначала по $n = |x|$ за полиномиальное время вычисляется некоторая булева схема S_n от n входных переменных, а затем ей на вход подаются биты слова x . Результат $S_n(x)$ оказывается правильным ответом на вопрос “ $x \in L$?”.

Языки $L \in BPP$ удается распознавать аналогичными, но вероятностными “схемными” алгоритмами. Разница состоит в том, что схема в этом случае имеет дополнительные входные переменные, $S_n = S_n(x, y)$, значения которых (т.е. y) надо задавать случайно, пользуясь датчиком. Тогда, с большой вероятностью, значение $S_n(x, y)$ совпадет с правильным ответом на вопрос “ $x \in L$?”.

Построить “схемный” разрешающий алгоритм для языка $L \in BPP$ можно из любого его частотного распознавателя $(R, q(n), \alpha(n))$. Достаточно в качестве $S_n(x, y)$ взять булеву схему, вычисляющую предикат $R(x, y)$ по битам слов x , $|x| = n$ и y , $|y| = q(n)$. Тогда $\alpha(n)$ будет вероятностью ошибочного ответа.

Следующая теорема показывает, что, в принципе, от ошибки можно избавиться вовсе с помощью хорошего частотного распознавателя и замены датчика случайности на специальный детерминированный выбор значений. Эффективный способ “улучшения” частотных распознавателей известен (лемма 1.3), но достаточно работоспособной методики выбора дополнительных битов нет.

Теорема 1.4 $BPP \subset P/Poly$.

Доказательство. Пусть $L \in BPP$. По лемме 1.3 для L существует частотный распознаватель вида $(R, q(n), (2\sqrt{2}/3)^{n^2})$. Обозначим через $Y(x)$ множество

$$\{y \mid |y| = q(|x|) \text{ и значение } R(x, y) \text{ противоположно } “x \in L”\}.$$

Доля множества $Y(x)$ среди всех слов y длины $q(n)$ не больше $(2\sqrt{2}/3)^{n^2}$ и

$$(2\sqrt{2}/3)^{n^2} \cdot 2^n < 1 \text{ при } n > n_0.$$

Отсюда для каждого $n > n_0$ найдется двоичное слово y_n такое, что

$$|y_n| = q(n), \quad y_n \notin \bigcup_{|x|=n} Y(x).$$

Для него выполняется

$$|x| = n > n_0 \Rightarrow (x \in L \Leftrightarrow R(x, y_n) = 1).$$

Пусть n достаточно велико. Так как $R \in P \subset P/Poly$, то существует булева схема размера $poly(|x| + |y|)$, вычисляющая предикат R по битам слова xy . При $|y| = q(|x|)$ размер этой схемы оценивается полиномом от длины x . Если взять $y = y_n$, где $n = |x|$, и добавить $q(n) + const$ операторов для вычисления битов слова y_n (оно одно для всех x длины n), то получится схема полиномиального размера, вычисляющая предикат “ $x \in L$ ” на словах длины n .

■

V. Krupski
COMPLEXITY
Lecture Notes, draft

Глава 2

Вероятностный алгоритм распознавания простых чисел.

2.1 Сведения из теории чисел.

Факт 1 (Следствие из Китайской теоремы об остатках) Пусть $n = u \cdot v$, $(u, v) = 1$. Тогда

$$(\mathbb{Z}_n, +, \cdot) \cong (\mathbb{Z}_u, +, \cdot) \oplus (\mathbb{Z}_v, +, \cdot)$$

и отображение $k \mapsto (k \bmod u, k \bmod v)$ есть соответствующий изоморфизм.

Факт 2 (Малая теорема Ферма) Если p – простое и не является делителем a , то $a^{p-1} \equiv 1 \pmod{p}$.

Факт 3 (Тривиальный) Если $b^2 \equiv 1 \pmod{p}$, а $b \not\equiv \pm 1 \pmod{p}$, то p – составное.

Доказательство. В кольце \mathbb{Z}_p имеем разложение 0 в произведение ненулевых сомножителей, $(b-1)(b+1) = b^2 - 1 = 0$, поэтому \mathbb{Z}_p не поле и p – составное. ■

2.2 Извлечение корней.

Лемма 2.1 *Задача поиска решения уравнения*

$$x^k = n, \quad k, n \in N,$$

в натуральных числах разрешима за полиномиальное время.

Доказательство. Решаем это уравнение приближенно методом половинного деления с точностью 1. Начальное приближение корня – отрезок $[a, b] := [1, n]$. При вычислении середины отрезка округляем до целого числа, т.е. $c := \lfloor (a + b)/2 \rfloor$. Для выбора очередного приближения (одного из отрезков $[a, c]$ или $[c, b]$ в зависимости от результата проверки условия “ $c^k < n$ ”) вычисляем c^k , пользуясь двоичным разложением числа k :

$$k = 2^{l_1} + \dots + 2^{l_t}, \quad c^k = c^{2^{l_1}} \cdot \dots \cdot c^{2^{l_t}},$$

а c^{2^l} вычисляем последовательным возведением в квадрат. При этом прерываем вычисление, если текущий результат стал больше n . Получится не более $(l_1 + \dots + l_t + t) = O(\log k)$ операций умножения чисел, не превосходящих n . Количество итераций, достаточных для получения приближения точности $(b-a) = 1$ есть $O(\log n)$. Итого, $O((\log k + \log n)^2)$ операций. Остается проверить, не является ли одно из чисел a, b точным решением, что можно сделать таким же образом за то же время.

Метод легко программируется на языке типа BASIC. Доказательство завершается стандартным моделированием этого языка машинами Тьюринга. ■

2.3 Вероятностный алгоритм распознавания простых чисел.

Вход: натуральное число $n > 2$.

Для получения правильного ответа с вероятностью ошибки $< 2^{-d}$ повторить следующую последовательность шагов d раз. Если ни разу не удалось установить, что число n – составное, то считать его простым.

1. Проверка n на четность (если четное, то составное).
2. Проверяем, что n не представимо в виде x^k для некоторых натуральных x и $k > 1$ (достаточно перебрать $k = 2, \dots, \lfloor \log_2 n \rfloor$). Если представимо, то оно составное.

3. Представляем четное число $n - 1$ в виде $2^k \cdot l$, где $k > 0$, а l – нечетно.
4. Выбираем a случайно из чисел $1, \dots, n - 1$.
5. Последовательно вычисляем $a^l, a^{2 \cdot l}, a^{2^2 \cdot l}, \dots, a^{n-1}$ по модулю n и ищем такое $j < k$, что

$$a^{2^j \cdot l} \not\equiv \pm 1 \pmod{n}, \quad a^{2^{j+1} \cdot l} \equiv 1 \pmod{n}.$$

Если нашли, то n – составное (Факт 3).

6. Предыдущий шаг закончился вычислением $a^{n-1} \pmod{n}$. Если $a^{n-1} \not\equiv 1 \pmod{n}$, то n – составное (Факт 2).

2.4 Верификация алгоритма.

Теорема 2.2 Если число $n > 2$ – простое, то алгоритм установит это. Если число n составное, то алгоритм установит это с вероятностью, не меньшей, чем $1 - 2^{-d}$.

Доказательство. Очевидно, что в случае простого n алгоритм никогда не объявит его составным. Неправильный ответ может получиться только в случае нечетного составного $n = u \cdot v$, $\text{нод}(u, v) = 1$. Покажем, что вероятность неправильного ответа в этом случае (т.е. объявления данного составного числа n простым) при одной итерации шагов 1 – 6 не превосходит $1/2$. Тогда после d итераций получим вероятность ошибки не более 2^{-d} .

Пусть \equiv означает сравнение по модулю n . Заметим, что если $\text{нод}(a, n) > 1$ для выбранного на шаге 4 числа a , то $a^{n-1} \not\equiv 1$, т.к. при $a = a_1 \cdot c$, $n = n_1 \cdot c$

$$a^{n-1} \equiv 1 \quad \Rightarrow \quad n_1 \equiv a_1^{n-1} \cdot c^{n-1} \cdot n_1 \equiv 0.$$

На шаге 6 это будет обнаружено и алгоритм даст правильный ответ. Поэтому достаточно установить, что по крайней мере для половины чисел из множества

$$G = \{a \mid 1 \leq a < n, \text{нод}(a, n) = 1\}$$

шаги 5, 6 также приведут к правильному ответу (т.е. обнаружат простоту n). Заметим, что G состоит в точности из всех обратимых элементов кольца \mathbb{Z}_n , т.е. является мультипликативной группой этого кольца.

Для абелевой группы (H, \cdot) обозначим

$$H^i = \{x^i \mid x \in H\}.$$

Тогда H^i – ее подгруппа, а отображение $x \mapsto x^i$ есть гомоморфизм H на H^i . При этом мощность множества $\{x \in H \mid x^i = h\}$ одна и та же для всех $h \in H^i$. Заметим также, что $H^i \supset (H^i)^2 = H^{2i}$.

Пусть U и V – мультипликативные группы колец вычетов \mathbb{Z}_u и \mathbb{Z}_v соответственно. Из Факта 1 следует, что $G \cong U \times V$ посредством отображения $\varphi(x) = (x \bmod u, x \bmod v)$. Рассмотрим убывающие последовательности подгрупп

$$\begin{aligned} G^l &\supset G^{2 \cdot l} \supset G^{2^2 \cdot l} \supset \dots \supset G^{2^k \cdot l} = G^{n-1} \supset \{1\}, \\ U^l &\supset U^{2 \cdot l} \supset U^{2^2 \cdot l} \supset \dots \supset U^{2^k \cdot l} = U^{n-1} \supset \{1\}, \\ V^l &\supset V^{2 \cdot l} \supset V^{2^2 \cdot l} \supset \dots \supset V^{2^k \cdot l} = V^{n-1} \supset \{1\}, \\ \varphi(G^{2^j \cdot l}) &= U^{2^j \cdot l} \times V^{2^j \cdot l}, \quad j = 0, \dots, k. \end{aligned}$$

При выборе каких $a \in G$ шаги 5, 6 не допускают ошибки, т.е. объявляют составное число $n = u \cdot v$ таковым? Это происходит, когда a есть решение одного из уравнений

$$\text{шаг 5: } x^{2^j \cdot l} \equiv g, \quad \text{где } g \in G^{2^j \cdot l}, \quad 0 \leq j < k, \quad (2.1)$$

$$g \not\equiv \pm 1, \quad g^2 \equiv 1,$$

$$\text{шаг 6: } x^{n-1} \equiv g, \quad \text{где } g \in G^{n-1}, \quad g \neq 1. \quad (2.2)$$

Сам набор уравнений зависит от n .

Случай 1: $U^{n-1} \neq \{1\}$ или $V^{n-1} \neq \{1\}$. Тогда $G^{n-1} \cong U^{n-1} \times V^{n-1}$ также содержит элементы, отличные от 1. Семейство множеств

$$M_g = \{x \in G \mid x^{n-1} = g\}, \quad g \in G^{n-1}$$

образует разбиение G на части по $|G|/|G^{n-1}|$ элементов каждая. Среди элементов $a \in G$ только элементы M_1 не удовлетворяют ни одному из уравнений (2.2). Частей не менее двух. Поэтому не менее половины элементов G удовлетворяют хотя бы одному из уравнений (2.2). Если на шаге 4 будет выбран один из них, то на шаге 6 число n будет объявлено составным (правильный ответ).

Случай 2: $U^{n-1} = V^{n-1} = \{1\}$. Заметим, что $U^l \neq \{1\}$ и $V^l \neq \{1\}$, так как l – нечетное и -1 принадлежит им обоим (под -1 понимается соответствующий вычет по модулю n , т.е. $-1 \equiv (n-1)$). Поэтому

$$j_0 = \min\{s \mid U^{2^s \cdot l} = V^{2^s \cdot l} = \{1\}\} - 1 \geq 0.$$

Пусть $t = 2^{j_0} \cdot l$. Множества G^t, U^t, V^t являются членами рассматриваемых последовательностей.

Подслучай 2.1: одно из множеств U^t, V^t есть $\{1\}$. В этом случае $-1 \notin G^t$, т.к. $(-1, -1) \notin U^t \times V^t$ (изоморфизм φ переводит G^t в $U^t \times V^t$, а $\varphi(-1) = (-1, -1)$). Рассуждаем аналогично случаю 1. Рассмотрим разбиение множества G на равномошные части

$$M_g = \{x \in G \mid x^t = g\}, \quad g \in G^t. \quad (2.3)$$

Частей не менее двух и только элементы одной из них (M_1) не удовлетворяют ни одному из уравнений (2.1) при $j = j_0$. Поэтому по крайней мере для половины элементов $a \in G$ верно, что если на шаге 4 будет выбран именно a , то на шаге 5 обнаружится, что $a^{2^{j_0} \cdot l} \not\equiv \pm 1$, $a^{2^{j_0+1} \cdot l} \equiv 1$, и число n будет объявлено составным.

Подслучай 2.2: $|U^t| > 1, |V^t| > 1$. Тогда $|G^t| = |U^t| \cdot |V^t| \geq 4$. Это означает, что в разбиении (2.3) частей не менее четырех и только элементы одной или двух из них (M_1 и может быть M_{-1}) не удовлетворяют ни одному из уравнений (2.1) при $j = j_0$. Далее повторяем те же рассуждения. ■

2.5 Оценка сложности.

Теорема 2.3 *Задача распознавания простых чисел принадлежит классу BPP.*¹

Доказательство. Оценим временную сложность вероятностного алгоритма распознавания простых чисел. Число итераций d считаем фиксированным. Достаточно показать, что шаги 1–6 моделируются вероятностной машиной Тьюринга за время, оцениваемое сверху полиномом от $\log n$. Это очевидно для всех шагов, кроме шага 4, где требуется вычислить значение случайной величины ξ с распределением

$$\text{Prob}\{\xi = a\} = 1/N, \quad a = 1, \dots, N$$

(при $N = n - 1$). Вероятностная машина Тьюринга снабжена лентой-датчиком случайных битов, что позволяет легко моделировать лишь распределения с двоично-рациональными вероятностями значений. Здесь же число N может быть произвольным.

¹См. сноску на стр. ??.

В предыдущем разделе мы показали, что алгоритм на входе n допускает ошибку только тогда, когда выбранное значение a принадлежит некоторому (определяемому по n) множеству M , причем $|M| \leq N/2$. Отсюда была получена оценка сверху вероятности ошибки $(\text{Prob}\{\xi \in M\})^d \leq (1/2)^d$. Если взять любое другое распределение случайной величины ξ , то оценка

$$(\text{Вероятность ошибки}) \leq (\text{Prob}\{\xi \in M\})^d$$

сохранится. Поэтому для доказательства теоремы достаточно научиться вычислять на вероятностной машине Тьюринга за время $\text{poly}(\log N)$ значение какой-нибудь случайной величины $\xi' \in \{1, \dots, N\}$ с $\text{Prob}\{\xi' \in M\} \leq p < 1$ (p не зависит от N), а затем выбрать в алгоритме параметр d из условия $p^d < 1/3$.

Алгоритм вычисления такой случайной величины ξ' :

Вход: число N . Выбираем число m из условия $2^{m-1} < N \leq 2^m$ и с помощью датчика случайных битов порождаем случайное двоичное слово v длины m . Пусть число k таково, что v есть двоичная запись числа $k - 1$ (с лидирующими нулями). Если $k \leq N$, то результатом объявляем k , а в противном случае — число 1.

Величина ξ' принимает значения из множества $\{1, \dots, N\}$, причем вероятность каждого значения не меньше 2^{-m} . Отсюда

$$\text{Prob}\{\xi' \in M\} \leq 1 - \frac{N}{2} \cdot 2^{-m} \leq 3/4.$$

Количество обращение к датчику случайных битов и общее время работы этого алгоритма есть $O(\log N)$. Если шаг 4 в алгоритме распознавания простых чисел реализовать таким образом, то d можно взять равным 4. ■