

Алгоритмы и модели вычислений.

Задание 3: Сложность вычислений, классы P, NP

Сергей Володин, 272 гр.

задано 2014.02.27

Задача 2

$f(n) = \text{poly}(n)$ — время работы машины M из условия на входе x длины n . За каждый такт машина читает не более одного символа, поэтому количество прочитанных символов $|y_r| \leq f(n)$. Причем машина не могла читать их не подряд, так как за один такт головка смещается на $\leq \pm 1$ ячейку.

1. Если $x \in L$, то, по условию, $\exists y: M(x, y) = 1$. Возьмем $y' = y[1...f(n)]$, тогда $|y'| = O(\text{poly}(|x|))$. Тогда $M(x, y') \equiv M(x, y)$, так как машина «не заметит» изменение длины слова (к суффиксу она не обращалась).
2. Если $\exists y' \in \Sigma^{f(|x|)}: M(x, y') = 1$, то возьмем $y = y'$, и по условию, $x \in L$.

Получаем $x \in L \Leftrightarrow \exists y' \in \Sigma^{f(|x|)}: M(x, y') = 1$. МТ полиномиальна по $|x|$, значит, полиномиальна по $|x\#y|$. Получаем $L \in \text{NP}$, в качестве полиномиального по $|x|$ сертификата берем $y'(x) = y(x)[1...f(|x|)]$, где $f(n)$ — полином из условия полиномиальности МТ по $|x|$.

(каноническое) Задача 8

Модифицируем доказательство из Кормена. Вход — массив A , $|A| = n$. Количество целиком заполненных групп из 9-ти элементов, медиана которых меньше x не меньше, чем $\lceil \frac{1}{2} \lceil \frac{n}{9} \rceil \rceil - 2$. Две группы не учитываются, так как в одной из них само x , а другая может не быть заполнена целиком. В каждой такой группе 5 элементов, не превышающих свою медиану, поэтому $A_{<x}$ — количество элементов A , меньших x будет $A_{<x} \geq 5 \lceil \frac{1}{2} \lceil \frac{n}{9} \rceil \rceil - 2 \geq \frac{5}{18}n - 10$. Аналогично $A_{>x} \geq \frac{5}{18}n - 10$. Тогда размеры групп $A_{\leq x}, A_{\geq x} \leq n - \frac{5}{18}n + 10 = \frac{13}{18}n + 10$. Рекуррентность:

$$T(n) = T(\lceil \frac{n}{9} \rceil) + T(\frac{13}{18}n + 10) + O(n).$$

Фиксируем n . Пусть $T(n') \leq cn$ для правой части (доказательство по дереву рекурсии, от листьев к корню). Пусть функция $O(n) \leq an$. Тогда $T(n) \leq c(\frac{n}{9} + 1 + \frac{13}{18}n + 10) + an = n(\frac{5}{6}c + a) + 11c$. Докажем, что эта величина также меньше cn (шаг индукции) при некоторых c : $n(\frac{5}{6}c + a - c) + 11c \leq 0 \Leftrightarrow n(a - \frac{c}{6}) + 11c \leq 0$. Возьмем $c = 7a$, откуда получим требуемое (при достаточно больших n неравенство выполнено).

(каноническое) Задача 11

$M_{p \times q}^{\mathbb{Z}, S}$ — множество матриц $\|a_{ij}\|$ размера $p \times q$ с целыми коэффициентами, такими, что $|a_{ij}| \leq S$. $S = 10000, m = 2014$. Язык $\{0, 1\}^* \supset L_n = \{\text{bin}(m, n, A, b) \mid m \in \mathbb{N}, (A, b) \in M_{m \times n}^{\mathbb{Z}, S} \times M_{m \times 1}^{\mathbb{Z}, S}, Ax = b \text{ — несовместна}\}$ — двоичные записи несовместных систем линейных уравнений с целыми коэффициентами (функция bin кодирует матрицу в двоичной записи).

1. Рассмотрим $w_j^i = (\|i \ 0 \ \dots \ 0\|, \|j\|)$. При $i = 0, j \in \{1, 2\}$ система несовместна, поэтому $\text{bin}(w_1^0), \text{bin}(w_2^0) \in L_{2014}$. При $i = 1, j \in \{1, 2\}$ система совместна, поэтому $\text{bin}(w_1^1), \text{bin}(w_2^1) \notin L_{2014}$.
2. (а) Опишем алгоритм и докажем его корректность. Рассмотрим расширенную матрицу $C = \|A|b|\|$. Модуль ее элементов не превосходит L . Будем применять к ней последовательно элементарные операции над строками S_i , получая матрицу $C'_i = \|A'_i|b'_i|\|$. Поскольку $Ax = b \Leftrightarrow A'_i x = b'_i$ (системы эквивалентны), исходная система совместна \Leftrightarrow полученная после операций система совместна. Применим метод Гаусса (прямой ход) к матрице C (ненулевые элементы берем не из последнего столбца), состоящий из элементарных операций над строками. Пусть в i -й строке найден столбец j с ненулевым элементом $a_{ij} \neq 0$. Перед методом Гаусса переставим строки так, чтобы $j'(i) = i$ (ненулевые элементы на главной диагонали) — элементарная операция над столбцами (т.е. переобозначим неизвестные). После прямого хода метода Гаусса получим матрицу

$$C' = \left\| \begin{array}{cccccc|c} 1 & & * & & & & b'_1 \\ & \ddots & & & & & \vdots \\ 0 & & 1 & & * & & b'_r \\ 0 & \dots & 0 & 0 & 0 & 0 & b'_{r+1} \\ & & & \dots & & & \\ 0 & \dots & 0 & 0 & 0 & 0 & b'_n \end{array} \right\|$$

Единицы получились именно на диагонали, так как столбцы были переставлены. r -я строка является последней ненулевой (в противном случае можно продолжить метод Гаусса)

(b) Докажем, что система несовместна $\Leftrightarrow \exists i \in \overline{r+1, n}: b'_i \neq 0$

i. \Leftarrow Имеем уравнение $0^T x = 1$

ii. \Rightarrow (от противного) Пусть система несовместна, и все b_i отличны от нуля. Выполним метод Гаусса до конца, убрав «*» выше единиц на диагонали. Левее столбца b' не могла получиться строка из нулей (по алгоритму вычитаем i -ю строку из всех строк выше, поэтому i -я единица на диагонали останется). Поэтому выше нет строк вида $\|0 \dots 0 \ 1\|$. Но их нет и ниже r -й строки, поэтому их нет вовсе. Метод Гаусса привел матрицу к упрощенному виду, и по Предложению 1 (Беклемишев, стр. 151) система совместна — противоречие.

(c) Рассмотрим метод Гаусса. Пусть $\{C_k\}_{k=0}^r$ — преобразованные матрицы, C_i — матрица после i шагов алгоритма (рассмотрены первые i строк). $C_0 \equiv C$. Обозначим элементы матрицы $A_k = \|a_k^{ij}\|$. Пусть алгоритм выполнил $k-1$ шагов. Рассмотрим изменение элементов матрицы на k -м шаге.

$$\left\| \begin{array}{cccc} \dots & a_{kk} & \dots & a_{kj} & \dots \\ & & & & \\ \dots & & & & \\ & & & & \\ \dots & a_{ik} & \dots & a_{ij} & \dots \end{array} \right\| \sim \left\| \begin{array}{cccc} \dots & 1 & \dots & \frac{a_{kj}}{a_{kk}} & \dots \\ & & & & \\ \dots & & & & \\ & & & & \\ \dots & a_{ik} & \dots & a_{ij} & \dots \end{array} \right\| \sim \left\| \begin{array}{cccc} \dots & 1 & \dots & \frac{a_{kj}}{a_{kk}} & \dots \\ & & & & \\ \dots & & & & \\ & & & & \\ \dots & 0 & \dots & a_{ij} - \frac{a_{kj}}{a_{kk}} a_{ik} & \dots \end{array} \right\|$$

i. k -я строка делится на a_{k-1}^{kk} , поэтому $a_k^{kj} = \frac{a_{k-1}^{kj}}{a_{k-1}^{kk}}$

ii. k -я строка вычитается из всех $k < i$ -х ниже

A. В k -м столбце нули ниже главной диагонали: $a_k^{ik} = 0, i > k$.

B. В $k < j$ -м столбце $k < i$ -й строки $a_k^{ij} = a_{k-1}^{ij} - a_{k-1}^{ik} \frac{a_{k-1}^{kj}}{a_{k-1}^{kk}}$.

«Вынесем за скобки» индекс $k-1$ (в этой формуле он один для всех a_{k-1}): $a_k^{ij} = \left(\frac{a_{k-1}^{ij} a_{k-1}^{kk} - a_{k-1}^{kj} a_{k-1}^{ik}}{a_{k-1}^{kk}} \right)_{k-1}$

Пусть дана матрица $A: m \times n$. Определим $\Delta_{j_1, \dots, j_t}^{i_1, \dots, i_t}$ — определитель подматрицы, полученной из A вычеркиванием всех строк кроме i_1, \dots, i_t и всех столбцов кроме j_1, \dots, j_t .

С этим обозначением $a_k^{ij} = \left(\frac{\Delta_{kj}^{ki}}{\Delta_k^{kk}} \right)_{k-1}$

$$\text{Получаем } A_k = \left\| \begin{array}{cccc} \dots & \dots & \dots & \dots \\ \dots & 1 & \dots & \left(\frac{a_{kj}^{kj}}{a_{kk}^{kk}} \right)_{k-1} \\ \dots & 0 & \dots & \dots \\ \dots & 0 & \dots & \left(\frac{\Delta_{kj}^{ki}}{\Delta_k^{kk}} \right)_{k-1} \\ \dots & 0 & \dots & \dots \end{array} \right\|, \text{ где «...» означают, что элементы не меняются.}$$

(d) (Задача 11.3)

i. (Я проверил для $k \leq 3$, т.е. утверждение не доказано). Получим по индукции формулу $a_k^{ij} = \frac{\Delta_{12\dots kj}^{12\dots ki}}{\Delta_{12\dots k}^{12\dots k}} ???$

ii. Из формулы выше следует, что получающиеся при промежуточных вычислениях числители и знаменатели элементов матрицы ограничены сверху $\max(|\Delta_{12\dots kj}^{12\dots ki}|, |\Delta_{12\dots k}^{12\dots k}|)$. По формуле полного разложения для числителя

$$\Delta_{12\dots kj}^{12\dots ki} = \sum_{t_1, \dots, t_{k+1}} (-1)^{\text{sign}(t_1, \dots, t_{k+1})} a_{xx} \cdot \dots \cdot a_{xx} \text{ (индексы опущены), что по модулю}$$

$$|\Delta_{12\dots kj}^{12\dots ki}| \leq [\max(m, n)]! \max_{A, b} |a_{ij}|^{\max(m, n)} \leq \boxed{\leq}. \text{ Обозначим } M = \max(m, n), \text{ получим } \boxed{\leq} M^M h^M = (Mh)^M. \text{ Аналогично для знаменателя.}$$

Итак, числители и знаменатели элементов матрицы, получающихся при промежуточных вычислениях, ограничены сверху $(Mh)^M$, где $M = \max(m, n)$.

(e) Для оценки времени работы приведен псевдокод:

```

1 //M[i][j] - matrix A/b
2 for(i = 1; i <= m; i++) // rows i=1...m
3 {
4     for(j = 1; j <= n; j++) // find j: aij != 0
5     {
6         if(M[i][j] != 0) // found
7         {
8             C = M[i][j];
9
10            // dividing i-th row by non-zero element
11            for(k = 1; k <= n + 1; k++)
12                M[i][k] /= C;
13
14            for(k = i + 1; k <= m; k++) // subtracting from row k down
15            {
16                C = M[k][j];
17                for(l = 1; l <= n + 1; l++) // column l
18                    M[k][l] -= M[i][l] * C;
19            }
20
21            break;
22        }
23    }
24 }
```

- (f) Храним в МТ рациональные числа как числитель и знаменатель. Оценим их сверху. Вернемся к формуле 2(c)iiB, запишем ее в виде $a_k^{ij} = \frac{\frac{a_1}{a_2} \frac{b_1}{b_2} - \frac{c_1}{c_2} \frac{d_1}{d_2}}{\frac{a_1}{a_2}} = \frac{a_1 b_1 c_2 d_2 - c_1 d_1 a_2 b_2}{b_2 c_2 d_2 a_1}$. Если числители и знаменатели на $k - 1$ шаге ограничены L , то на $k + 1$ -м они будут ограничены $2L^4$. Рассуждая по индукции, на последнем шаге получим, что они ограничены $2(\dots 2(2L^4)^4)\dots)^4$, где возведение в четвертую степень происходит количество раз, равное рангу матрицы (количество шагов алгоритма). Но он не превосходит $n = 2014$. Поэтому максимальный модуль числа фиксирован. Получаем, что арифметические операции выполняются за $O(1)$.
- (g) Оценим время работы как $T(A, b, m) \leq m \times n \times (O(1) + n \times O(1) + m \times (O(1) + n \times O(1))) \stackrel{n=2014}{=} O(m^2)$. $\text{bin}(\cdot)$ — двоичная запись числа. Длина входа $I(A, b, m) = (mn + m) \min_{A, b} |\text{bin}(a_{ij})| = \Omega(m) \geq cm$, поэтому $T(A, b, m) \leq c_1 m^2 \leq c I^2(A, b, m) = O(I^2)$.

(каноническое) Задача 12

- (a) Используем быстрое возведение в степень по модулю d . Умножаем числа не более, чем по $2|d|$ бит. Остаток от деления считается за квадрат длины битовой записи. Псевдокод:

```
1 number power(a, b, d)
2 {
3     if(b == 0) return(1);
4     if(b % 2 == 0)
5     {
6         number x = power(a, b / 2, d);
7         return((x * x) % d);
8     }
9     else
10    {
11        number x = power(a, (b - 1) / 2, d);
12        x = (x * x) % d;
13        return((a * x) % d);
14    }
15 }
16 ans = (power(a, b, d) == (c % d));
```

На каждом шаге второй аргумент уменьшается как минимум вдвое, поэтому высота дерева рекурсии $h \leq \log_2 b$. На каждом шаге производятся операции над числами битовой длины не более $2 \log d$, на листе дерева рекурсии ($b = 0$) выполняется $O(1)$ операций. Последний шаг (сравнение) выполняется за $O(\log d)$ операций. Сложность арифметических операций не более, чем квадратичная по длине битовой записи.

Получаем $T(a, b, c, d) \leq \log_2 b \cdot O(\log^2 d) + O(1) = O(\log^2 d \log b)$. Длина входа $I(a, b, c, d) = \log a + \log b + \log c + \log d$, поэтому $T = O(I^3)$.

- (b) Слова, соответствующие $(1, 1, 1, 2)$, $(1, 2, 1, 2) \in L$, $(1, 1, 2, 2)$, $(1, 2, 2, 2) \notin L$

(каноническое) Задача 13

Бинарным ищем корень 2014 степени. $L = 1$, R — вход. Шагов $\log_2 R = \log_2 2^t = t$, возводим числа $\leq 2^t$ в 2014 степень за $\log^{2014} 2^t = t^{2014}$. Псевдокод:

```
1 number L = 1;
2 number R = X = input();
3
4 number M, B = 2014;
5 while(R - L > 1)
6 {
7     M = (R + L) / 2;
8     if(power(M, B) < X)
9         R = M;
10    else L = M;
11 }
12 ans = 0;
13 if(power(L, B) == X)
14     ans = 1;
15 else if(power(R, B) == X)
16     ans = 1;
```

Поддерживается свойство: ответ всегда лежит в $[L, R]$. На каждой итерации цикла $|R - L|$ уменьшается вдвое, откуда цикл совершает $O(\log X)$ итераций. На каждой производится возведение в степень $B = 2014$ за $O(\log^{2014} X)$. Последние сравнения занимают $O(\log^{2014} X)$, поэтому $T(I) = O(\log X) \cdot O(\log^{2014} X) + O(\log^{2014} X) = O(\log^{2015} X)$, где длина входа I — длина битовой записи числа X , т.е. $I = \Theta(\log X)$, откуда $T = O(I^{2015})$.

(каноническое) Задача 14

Определение замкнутости P относительно \cdot^* :

$$\forall L \in P \hookrightarrow L^* \in P.$$

Пусть $L \in P$. Тогда \exists МТ M : время ее работы $T(|w|) = f(|w|) = \text{poly}(|w|)$. Построим другую МТ M' , которая будет разными способами разбивать входное слово $w \in L^*$ на подслова $w = w_n \dots w_1$, и проверять $w_i \in L$ за полиномиальное время по $|w_i|$. Перебираем правые границы слов, запоминаем те, в которых могут заканчиваться слова из L . Левые границы берем из множества правых границ $(+1)$. Если правая граница n — слово найдено.

Всего подслов не больше, чем $|w|^2$ (выбираем левую и правую границу), поэтому время работы МТ M' $T'(|w|) \leq |w|^2 \max f(|w_i|) \leq |w|^2 \text{poly}(|w|) = \text{poly}(|w|) \Rightarrow L^* \in P$.

(каноническое) Задача 15

(не дописано)

1. DA
 - (a) $DA, L(\cdot) = \emptyset$. Обходом графа в ширину ищем пути из принимающего состояния. Время $T = O(|V| + |E|)$, где $|V|$ и $|E|$ — количества вершин и ребер соответственно. Длина входа I — описание графа. $I = \Theta(|V|^2)$ (матрица смежности). $|E| \leq |V|^2$, поэтому $T = O(|V|^2) = O(I)$.
 - (b) $DA, |L(\cdot)| = \infty$. Ищем циклы в графе обходом в ширину.
 - (c) $DA, w \in L(\cdot)$. Переходим по графу за $O(|w|)$. Если перешли в принимающее состояние автомата — МТ переходит в $q \in Acc$. МТ останавливается в любом случае, так как для каждого символа слова совершается один переход в автомате за ограниченное сверху время.
 - (d) $DA, w \notin L(\cdot)$. Решаем предыдущую разрешимую задачу и выдаем противоположный ответ.
2. NA
 - (a) Работает тот же алгоритм, что и для DA .
 - (b) Работает тот же алгоритм, что и для DA .
 - (c) Храним не одно состояние автомата, а множество состояний, в котором он может оказаться при прочтении префикса слова. Поддерживаем это свойство для каждого нового символа. В конце, если среди множества есть принимающие состояния автомата, МТ переходит в принимающее состояние.
 - (d) Предыдущая задача, противоположный ответ.
3. R . Строим НКА за линейное по размеру R время. Далее аналогично.
4. G .
5. \mathcal{A}, \mathcal{B} — ДКА. Построим минимальные ДКА за полиномиальное по $|A| + |B|$ время: на каждом шаге алгоритма количество состояний уменьшается, поэтому количество шагов не превосходит $|\mathcal{A}|$. На каждом шаге выполняется полиномиальное число действий (от количества состояний). Проверим изоморфность двух минимальных ДКА за $|A| + |B|$. Длина входа $|A|^2 + |B|^2$ (графы входных автоматов заданы матрицами смежности).