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 $|y_r| \leqslant f(n)$. Причем машина не могла читать их не подряд, так как за один такт головка смещается на $\leqslant \pm 1$ ячеек.

1. Если $x \in L$, то, по условию, $\exists y\colon M(x,y) = 1$. 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 $\exists y' \in \Sigma^{f(|x|)}\colon M(x,y') = 1$, то возьмем $y = y'$, и по условию, $x \in L$.

Получаем $x \in L \Leftrightarrow \exists y' \in \Sigma^{f(|x|)}\colon M(x,y') = 1$. МТ полиномиальна по $|x|$, значит, полиномиальна по $|x\#y|$. Получаем $L \in \mathsf{NP}$, 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 11

$\mathrm{M}_{p\times q}^{\mathbb{Z},S}$ — множество матриц $||a_{ij}||$ размера $p \times q$ с целыми коэффициентами, такими, что $|a_{ij}| \leqslant S$. $S = 10000, m = 2014$. Язык $\{0,1\}^* \supset L_{m\times n} = \{\text{bin}(m,n,A,b) \big| (A,b) \in \mathrm{M}_{m\times n}^{\mathbb{Z},S} \times \mathrm{M}_{m\times 1}^{\mathbb{Z},S},\ Ax = b \text{ — несовместна}\}$ — двоичные записи несовместных систем линейных уравнений с целыми коэффициентами.

1. Рассмотрим $w_j^i = \left(\begin{Vmatrix} i & 0 & ... & 0 \end{Vmatrix}, \begin{Vmatrix} j \end{Vmatrix} \right)$. При $i = 0$, $j \in \{1,2\}$ система несовместна, поэтому $w_1^0,\ w_2^0 \in L_{2014\times 1}$. При $i = 1$, $j \in \{1,2\}$ система совместна, поэтому $w_1^1,\ w_2^1 \notin L_{2014\times 1}$

2. (a) Опишем алгоритм и докажем его корректность. Рассмотрим расширенную матрицу $C = \big|\big|A\big|b\big|\big|^{\square}$. Модуль ее элементов не превосходит L. Будем применять к ней последовательно элементарные операции над строками S_i, получая матрицу $C_i' = \big|\big|A_i'\big|b_i'\big|\big|^{\square}$. 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

$$C' = \begin{Vmatrix} 1 & & \ast & & & & b_1' \\ & \ddots & & & & \ast & \vdots \\ \mathbf{0} & & 1 & & & & b_r' \\ 0 & ... & 0 & 0 & 0 & 0 & b_{r+1}' \\ & & & ... & & & \\ 0 & ... & 0 & 0 & 0 & 0 & b_n' \end{Vmatrix}$$

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

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

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

   ii. $\boxed{\Rightarrow}$ (от противного) Пусть система несовместна, и все b_i отличны от нуля. Выполним метод Гаусса до конца, убрав «$\ast$» выше единиц на диагонали. Левее столбца b' не могла получится 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 $\begin{Vmatrix} 0 & ... & 0 & 1 \end{Vmatrix}$. 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 $A_k = \big\|a_k^{ij}\big\|$. Пусть алгоритм выполнил $k-1$ шагов. Рассмотрим изменение элементов матрицы на k-м шаге.

$$\begin{Vmatrix} ... & a_{kk} & ... & a_{kj} & ... \\ & & ... & & \\ ... & a_{ik} & ... & a_{ij} & ... \end{Vmatrix} \sim \begin{Vmatrix} ... & 1 & ... & \frac{a_{kj}}{a_{kk}} & ... \\ & & ... & & \\ ... & a_{ik} & ... & a_{ij} & ... \end{Vmatrix} \sim \begin{Vmatrix} ... & 1 & ... & \frac{a_{kj}}{a_{kk}} & ... \\ & & ... & & \\ ... & 0 & ... & a_{ij} - \frac{a_{kj}}{a_{kk}} a_{ik} & ... \end{Vmatrix}$$

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

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

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

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

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

Пусть дана матрица $A : m \times n$. Определим $\Delta^{i_1,\dots,i_t}_{j_1,\dots,j_t}$ — 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 $a^{ij}_k = \left(\frac{\Delta^{ki}_{kj}}{\Delta^k_k}\right)_{k-1}$

Получаем $A_k = \left\| \begin{matrix} \dots & \dots & \dots & \dots & \dots \\ \dots & 1 & & \left(\frac{a^{kj}}{a^{kk}}\right)_{k-1} & \\ \dots & 0 & & & \\ \dots & 0 & & \left(\frac{\Delta^{ki}_{kj}}{\Delta^k_k}\right)_{k-1} & \\ \dots & 0 & & & \end{matrix} \right\|$, где «...» означают, что элементы не меняются.

(d) *(Я проверил для $k <= 3$, т.е. утвверждение не доказано).* Получим по индукции формулу $a^{ij}_k = \frac{\Delta^{12\dots ki}_{12\dots kj}}{\Delta^{12\dots k}_{12\dots k}}$???

(e) Из формулы выше следует, что получающиеся при промежуточных вычислениях числители и знаменатили элементов матрицы ограничены сверху $\max(|\Delta^{12\dots ki}_{12\dots kj}|, |\Delta^{12\dots k}_{12\dots k}|)$. По формуле полного разложения для числителя $\Delta^{12\dots ki}_{12\dots kj} = \sum\limits_{t_1,\dots,t_{k+1}} (-1)^{\mathrm{sign}(t_1,\dots,t_{k+1})} a_{xx} \cdot \dots \cdot a_{xx}$ (индексы опущены), что по модулю $|\Delta^{12\dots ki}_{12\dots kj}| \leqslant [\max(m,n)]! \max\limits_{A,b} |a_{ij}|^{\max(m,n)} \boxed{\leqslant}$. Обозначим $M = \max(m,n)$, получим $\boxed{\leqslant} M^M L^M = (ML)^M$. Аналогично для знаменателя.
Итак, числа, получающиеся при промежуточных вычислениях, ограничены $(ML)^M$, что обозначим за X. (Задача 11.3)

(f) Для оценки времени работы приведен псевдокод:

```
//M[i][j] - matrix A|b
for(i = 1; i <= m; i++) // rows i=1...m
{
  for(j = 1; j <= n; j++) // find j: aij != 0
  {
    if(M[i][j] != 0) // found
    {
      C = M[i][j];

      // dividing i-th row by non-zero element
      for(k = 1; i <= n + 1; i++)
        M[i][k] /= C;

      for(k = i + 1; k <= m; k++) // subtracting from row k down
      {
        C = M[k][j];
        for(l = 1; l <= n + 1; l++) // column l
          M[k][l] -= M[i][l] * C;
      }

      break;
    }
  }
}
```

(g) Храним в МТ рациональные числа как числитель и знаменатель. Оценим их сверху. Вернемся к формуле 2(c)iiB, запишем ее в виде $a^{ij}_k = \frac{\frac{a_1}{a_2}\frac{b_1}{b_2} - \frac{c_1}{c_2}\frac{d_1}{d_2}}{\frac{a_1}{a_2}} = \frac{a_1b_1c_2d_2 - c_1d_1a_2b_2}{b_2c_2d_2a_1}$. Если числители и знаменатели на $k-1$ шаге ограничены L, то на $k+1$-м они будут ограничены L^4. Рассуждая по индукции, на последнем шаге получим, что они ограничены $(\dots((L^4)^4)\dots)^4$, 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 $O(1)$.

(h) Оценим время работы как $T(A,b,m) = m \times n \times (n + O(1) + m \times (O(1) + n \times O(1))) = O(m^2)$. Длина входа $I(A,b,m) = (mn + m)\min\limits_{A,b} a_{ij} = \Omega(m) \geqslant cm$, поэтому $T(A,b,m) \leqslant c_1 m^2 \leqslant cI^2(A,b,m) = O(I^2)$.

(каноническое) Задача 12

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

```
number power(a, b, d)
{
   if(b == 0) return(1);
   if(b % 2 == 0)
   {
      number x = power(a, b / 2, d);
      return((x * x) % d);
   }
   else
   {
      number x = power(a, (b - 1) / 2, d);
      x = (x * x) % d;
      return((a * x) % d);
   }
}
ans = (power(a, b, d) == (c % d));
```

На каждом шаге второй аргумент уменьшается как минимум вдвое, поэтому высота дерева рекурсии $h \leqslant \log_2 b$. 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 $T(a, b, c, d) \leqslant \log_2 b \cdot O(\log^2 d) + O(1) = O(\log^2 d \log b)$. Длина входа $I(a, b, c, d) = \log a + \log b + \log c + \log d$, поэтому $T = O(I^3)$.

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

(каноническое) Задача 13

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

```
number L = 1;
number R = X = input();

number M, B = 2014;
while(R - L > 1)
{
  M = (R + L) / 2;
  if(power(M, B) < X)
    R = M;
  else L = M;
}
```

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 $O(\log^{2014} X)$. Поэтому $T(I) = O(\log X)$, где длина входа I — длина битовой записи числа X, т.е. $I = \Theta(\log X)$, откуда $T = O(I^{2015})$.

(каноническое) Задача 14

(каноническое) Задача 15

1. DA

   (a) $DA, L(\cdot) = \varnothing$. 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). $|E| \leqslant |V|^2$, 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. $\mathcal{A}, \mathcal{B}$ — 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).