Домашнее задание №7

Дедлайн: 01 апреля 2019 г., 23:00

Основные задачи

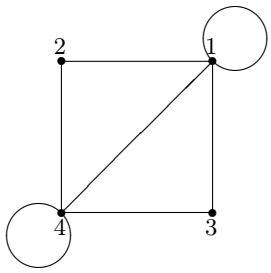
1. (2 + 2 + 2 балла) Рассмотрим такую задачу: на вход подаются 3 целочисленных матрицы A, B, C размеров $n \times n$; нужно проверить, выполнено ли равенство $C = AB$. Можно решить такую задачу просто перемножив матрицы A и B за $\Theta(n^3)$ арифметических операций стандартным способом (либо за $\Theta(n^{\log_2 7})$ при помощи алгоритма Штрассена перемножения матриц) с целыми числами, длина записи которых составляет $O(\log(nh))$, где h — самое большое абсолютное значение элементов матриц A и B. Вместо этого будем проверять равенство $C = AB$ рандомизированно. Сгенерируем случайный вектор $x \in \mathbb{R}^n$, компоненты которого равномерно распределены на $\{0, 1, 2, \ldots, N-1\}$. Затем проверим равенство $Cx = A(Bx)$. Чтобы его проверить, нам нужно 3 раза умножить матрицу на вектор, то есть нужно произвести $O(n^2)$ арифметических операций с числами, длина записи которых составляет $O(\log(nh))$, где h — самое большое абсолютное значение элементов матриц A, B, C и вектора x. Если равенство не выполнено, то сигнализируем об ошибке (выдаём 0), в противном случае выдаём 1, что означает, что исходное матричное равенство либо верно, либо мы подобрали плохой вектор x.

   (i) Принадлежность к какому классу сложности показывает описанный алгоритм?

   (ii) Оцените вероятность того, что $C \neq AB$ в случае $Cx = A(Bx)$

   (iii) Каким нужно выбрать N, чтобы гарантировать вероятность ошибки меньше p?

2. (1+1+1+1 балл) Диаграмма графа G изображена на рисунке.



   Путь в графе — это произвольная последовательность смежных вершин (возможны возвраты): $c = \{v_1, \ldots, v_l\}$. По определению, длина пути c равна $l-1$. Пусть g_n — это число путей в G длины n, которые начинаются в вершине 1. Из определения следует, что $g(0) = 1$ (единственный путь: $0 \to 0$), а $g(1) = 4$ (пути: $1 \to 1$, $1 \to 2$, $1 \to 4$, $1 \to 3$). Пусть A — это матрица инциденций графа G:

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

   (i) Вычислите число $g(2)$ путей в G длины 2 и проверьте, что оно равно сумме элементов первой строки матрицы A^2. Объясните это совпадение и докажите общую формулу для $g(n)$.

   (ii) Найдите рекуррентное соотношение, которому удовлетворяет последовательность $\{g_n,\, n = 0, 1, \ldots\}$. *Подсказка. В ответе должна получиться рекуррентность с целыми коэффициентами типа рекуррентности Фибоначчи: $g_{n+2} = Pg_{n+1} + Qg_n$. Можно просто подобрать коэффициенты и* **доказать**, *что они искомые. При этом необходимо вычислить хотя бы еще одно значение $g(n)$.*

Рассмотрим два способа вычисления $\{g_n,\ n = 0, 1, \ldots\}$.

Первый, основан на том, что последовательность $\{g_n\}$ удовлетворяет рекуррентному соотношению, т. е. разностному уравнению, и это подсказывает следующий *матричный* способ ее вычисления. Имеет место матричная формула[1] $\begin{pmatrix} g_n \\ g_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ Q & P \end{pmatrix}^{n-1} \begin{pmatrix} g_1 \\ g_2 \end{pmatrix}$

При вычислении $\{g_n\}$ этим способом можно использовать быстрое, например, "индийское возведение в степень" n за $O(\log n)$ тактов[2].

Второй способ вычислений основан на явном аналитическом решении линейной рекуррентности, которое можно получить самостоятельно или воспользоваться алгоритмом из текста на сайте. Например, для чисел Фибоначчи ($F_1 = 1, F_2 = 1, \ldots F_{n+2} = F_{n+1} + F_n$)) этот способ приводит к известной формуле Бине: $F_n = \frac{(\frac{1+\sqrt{5}}{2})^n - (\frac{1-\sqrt{5}}{2})^n}{\sqrt{5}}$.

У вас должна получится похожая формула, также содержащая квадратичную иррациональность $\sqrt{d} = \sqrt{P^2 + 4Q}$. Вычисление по аналитической формуле реализуется чуть хитрее, чем по матричной. Для каждого значения n нужно специфицировать операции и указать с какой точностью их нужно проводить (например, для вычисления $\sqrt{d}$ нужно указать процедуру вычисления, задать точность вычисления и оценить битовую трудоемкость процедуры).

На самом деле, переход к собственным векторам матрицы $\begin{pmatrix} 0 & 1 \\ Q & P \end{pmatrix}$ показывает, что оба алгоритмы тесно связаны, и матричный алгоритм можно рассматривать как корректный способ округления ответа, полученного по аналитической формуле.

Оценим трудоемкость нескольких алгоритмов вычисления g_n по простому модулю p.

(iii) Непосредственное вычисление по рекуррентной формуле. Оцените его трудоемкость при вычислении $A = g_{20000} \pmod{29}$.

(iv) Докажите, что последовательность $\{g_n\}$ периодична по любому модулю. Оцените ее период для $\pmod{29}$ и найдите трудоемкость вычисления (сложность нахождения периода + сложность вычисления A) этим способом.

3. (1+1+1 балл)

(i) Пусть НОД$(a, N) = 1$ и $a^{N-1} \neq 1 \pmod{N}$. Докажите, что по крайней мере для половины чисел из промежутка $1 \leq b < N$ выполнено $b^{N-1} \neq 1 \pmod{N}$.

(ii) Покажите, что Тест Ферма может быть реализован за полиномиально по входу число операций.

(iii) Покажите, что если N не является числом Кармайкла, то Тест Ферма даёт правильный ответ с вероятностью $\geqslant \frac{1}{2}$.

4. (1 балл) Пусть вычет a имеет показатель (порядок) δ_1 по $\pmod{n_1}$ и показатель n_2 по $\pmod{n_2}$, причём n_1 и n_2 взаимно просты. Найдите показатель a по $\pmod{n_1 n_2}$

Дополнительные задачи

1. (1 + 2 + 2 балла) Рассмотрим двудольный граф, в каждой доле которого по n вершин. Научимся рандомизированно проверять, есть ли в нём полное паросочетание. Сопоставим этому графу матрицу, такую что a_{ij} равно 1 если между вершиной i первой доли и вершиной j второй доли есть ребро и 0 в противном случае. Такая матрица называется матрицей Эдмондса. Её перманент равен количеству полных паросочетаний в графе (это один из пунктов задачи). Но задача вычисления перманента очень сложна. Тем не менее если нам нужно только проверить, что перманент не равен 0 мы можем использовать следующий подход: заменим a_{ij} равные единице на независимые переменные x_{ij}, а остальные оставим нулями. Тогда перманент матрицы равен нулю тогда и только тогда когда определитель матрицы как многочлен от x_{ij} равен 0 тождественно. Поэтому мы можем подставлять вместо x_{ij} случайные числа из $\{0, 1, \ldots, N-1\}$ и считать многочлен в этих точках.

(i) Докажите, что перманент матрицы Эдмондса равен количеству полных паросочетаний в графе.

(ii) Докажите, что перманент равен 0 $\Longleftrightarrow$ детерминант как многочлен от x_{ij} равен 0.

(iii) Оцените вероятность ошибки алгоритма и битовую сложность его работы.

2. (2+2+2 балла)

[1] Поскольку для $\{g_n\}$ коэффициенты P и Q — целые, то при вычислениях можно использовать только целую арифметику.

[2] Мы разбирали этот алгоритм в первом задании, и он с необходимыми изменениями переносится на матрицы.

(i) Пусть известно, что 5 *является квадратичным вычетом* по $\pmod p$, например, $p = 29$, т. е. разрешимо уравнение $x^2 = 5 \pmod p$. Обоснуйте алгоритм непосредственного вычисления A по аналитической формуле, т. е. прямо извлекая квадратный корень в конечном поле $\pmod p$ и проводя дальнейшие арифметические вычисления. Вычислите A этим способом. Оцените трудоемкость вычисления $g_n \pmod p$ для этого случая.

(ii) Пусть теперь 5 *НЕ является квадратичным вычетом* по $\pmod p$, например, $p = 23$. Придумайте и обоснуйте использующий аналитическую формулу алгоритм вычисления чисел $\{g_n\}$ по такому модулю. Проведите вычисления $A = g_{10000} \pmod{23}$. Оцените трудоемкость вычисления $g_n \pmod p$ для этого случая.

В этой задаче вам предстоит разобраться, как использовать матричный алгоритм для вычислении реккурентности по простому модулю. Мы уже знаем, что эффективность процедуры сильно (или даже критически) зависит от вычисления периода $\{g_n\} \pmod p$. И, собственно, основной вопрос, на который хочется найти ответ, как найти период в матричном представлении $\{g_n\}$. Предлагается придумать алгоритм самостоятельно и/или проанализировать следующий способ. Заметим, что нам повезло, и матрицу $\left(\begin{smallmatrix} 0 & 1 \\ Q & P \end{smallmatrix}\right)$ можно диагонализировать, т. е. привести ее к виду $S\left(\begin{smallmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{smallmatrix}\right)S^{-1}$, где λ_1, λ_2 — это собственные числа, а S — невырожденная матрица. Возводя в n-ю степень, получим $S\left(\begin{smallmatrix} \lambda_1^n & 0 \\ 0 & \lambda_2^n \end{smallmatrix}\right)S^{-1}$.

(iii) Наша задача состоит в том, чтобы обосновать эти манипуляции при вычислениях $\pmod p$ и понять, в какую степень нужно возвести матрицу, чтобы получилась единичная матрица, т. е. получить аналог малой теоремы Ферма для матриц указанного вида[3]

Оцените сложность вычисления периода и вычисления $\{g_n\} \pmod p$ матричным способом и сравните его с алгоритмами из пунктов (i)–(ii).

Комментарий. В последней задаче в пунктах (i)–(ii) речь идет о т.н. *квадратичных вычетах* и возникает естественный вопрос, как по данному числу a проверить разрешимость уравнения $x^2 = a \pmod p$ (в задаче $a = 5$, $n = 29, 23$). Можно, конечно, перебрать все вычеты, используя $O(p)$ операций, но есть и более интеллигентный полиномиальный по **длине записи** $\log p$ алгоритм, Он основан на знаменитом *квадратичном законе взаимности*, о котором можно прочитать в книге **[Виноградов, гл. 2]** (и придумать алгоритм самостоятельно). Но есть и еще более простой способ, основанной на обобщении малой теоремы Ферма. Если определить (см. , **[Виноградов, гл. 2]**) т. н. *символ Лежандра*: $\left(\frac{a}{p}\right)$, равный $+1$, если число $a \neq 0 \pmod p$ является квадратичным вычетом $\pmod p$, и, равный -1, в противном случае, то имеет место равенство $a^{\frac{p-1}{2}} = \left(\frac{a}{p}\right) \pmod p$. Таким образом, можно эффективно проверить, является ли a квадратичным вычетом, используя быстрое возведение в степень. Кроме того, можно построить быстрый вероятностный алгоритм поиска квадратичного вычета или невычета. Что понимается по этим, поясняет задача 3 (i).

3. (1+ 2 балла) Рассмотрим систему RSA.

(i) Пусть открытый ключ Боба $(25, 2021)$. Он хочет послать сообщение (число), используя протокол RSA. В какую степень он должен его возвести?

(ii) Докажите или опровергните, что кодирование в системе RSA $M \to M^e \mod n$ биективно отображает множество $\{0, 1, \ldots, n-1\}$ в себя.

4. (2 балла) Предложите алгоритм вычисления символа Лежандра по простому модулю, битовая сложность которого равна $O(\log^2 n)$. Ответ обосновать.

5. (1 + 1 балл) Докажите теоремы 15 и 16 из конспекта шестого семинара.

6. (1 + 3 балла) Докажите теоремы 17 и 18 из конспекта шестого семинара.

[3] Обратите внимания, что прямого аналога малой теоремы Ферма для матриц быть не может, поскольку, например, существуют т.н. нильпотентные матрицы (какая-то их степень равна нулевой матрице). Удивительно, но тексты, посвященные аналогам теоремы Ферма для матриц, появились только в этом тысячелетии (см., www.mathnet.ru/mp238). В принципе, их мог написать сильный студент.