

SPb HSE, ПАДИИ, 1 курс, весна 2024/25

Практика по алгоритмам #30

Фурье

22 мая

Собрано 23 мая 2025 г. в 21:47

Содержание

1. Фурье	1
2. Разбор задач практики	3
3. Домашнее задание	5
3.1. Обязательная часть	5
3.2. Дополнительная часть	5

Фурье

1. Рюкзак

Решите задачу о рюкзаке (subsetsum) с помощью умножения многочленов.

2. Билетики

Количество счастливых билетов из $2n$ цифр по модулю m за $\mathcal{O}(n \log n)$.

3. Возведение в степень

За какое время можно посчитать 2^n в десятичной системе счисления?

(*) Подумайте про выбор системы счисления и реальное время работы.

4. Рекуррентное соотношение

Проверить для последовательности $f_0, f_1, \dots, f_n$ и коэффициентов $a_1, \dots, a_k$, правда ли $\forall i \geq k \ f_i = \sum_j f_{i-j} a_j$. $\mathcal{O}(n \log n)$.

5. Поиск с ошибками

Даны текст t и строка s над алфавитом размера k . Для каждого из $|t| - |s| + 1$ наложений s на t узнать количество ошибок. Время $\mathcal{O}(k \cdot |t| \log |t|)$.

6. Поиск с ошибками и шаблоном

Апгрейд предыдущей задачи. И в тексте, и в строке допустимы символы «?».

7. Дуэль!

В каждой клетке полоски $1 \times n$ или растёт дерево, или нет. За $\mathcal{O}(n \log n)$ найдите количество троек деревьев, подходящих для дуэли (два дуэлянта и секундант). Тройка деревьев на позициях $i < j < k$ подходит, если $j - i = k - j$.

8. 3-SUM

Формулировка 3-SUM: дан массив A , выбрать $a, b, c \in A$: $a + b + c = 0$. В общем виде задача не решается за $\mathcal{O}(n^{2-\epsilon})$, но, если $A \subseteq \mathbb{Z} \cap [-C, C]$, есть простое решение за $\mathcal{O}(C \log C)$.

9. Уравнение

Даны n и m . Найти число троек (x, y, z) : $x^n + y^n \equiv z^n \pmod m$.

(*) Ускорьте предподсчёт, чтобы при $n = \Theta(m)$ работал за $\Theta(n)$.

10. Цепочка умножений

Хотим посчитать $P_1(x) \cdot \dots \cdot P_k(x)$, все многочлены степени n . За сколько можно это сделать? Почему нельзя сделать k прямых FFT и 1 обратное?

11. FFT по простому модулю

Мы прошли FFT над полем комплексных чисел. Предложите аналог над полем остатков по простому модулю. Примеры простых: $2^{18} \cdot 3 + 1$, $2^{20} \cdot 7 + 1$, $2^{25} \cdot 5 + 1$.

12. (*) FFT и повышение точности

Перемножьте с помощью FFT $A, B \in K[\mathbb{Z}/10^9\mathbb{Z}]$, степени многочленов до 10^6 .

Есть два решения: используя FFT над $\mathbb{C}$ и FFT над $\mathbb{Z}/p\mathbb{Z}$.

13. (*) Поиск подстроки в строке

За $\mathcal{O}(1)$ вызовов FFT найти подстроку в строке за $\mathcal{O}(n \log n)$. Алфавит произвольный.

14. (*) Inplace FFT

Ниже — код оптимизированного FFT. Прочитайте код. Поймите. Как изменить FFT, чтобы она стала inplace (изменяла сразу массив **a**)?

```

1  const int K = 20, N = 1 << K;
2  complex<double> root[N];
3  int rev[N];
4  void init():
5      for (int j = 0; j < N; j++)
6          rev[j] = (rev[j >> 1] >> 1) + ((j & 1) << (K - 1));
7      for (int k = 1; k < N; k *= 2)
8          num tmp = exp(PI / k);
9          root[k] = {1, 0}; // в root[k..2k) хранятся первые k корней степени 2k
10         for (int i = 1; i < k; i++)
11             root[k+i] = (i & 1) ? root[(k+i) >> 1] * tmp : root[(k+i) >> 1];
12 void FFT(a, fa): // a → fa
13     for (int i = 0; i < N; i++)
14         fa[rev[i]] = a[i];
15     for (int k = 1; k < N; k *= 2)
16         for (int i = 0; i < N; i += 2 * k)
17             for (int j = 0; j < k; j++)
18                 num tmp = root[k + j] * fa[i + j + k];
19                 fa[i + j + k] = fa[i + j] - tmp;
20                 fa[i + j] = fa[i + j] + tmp;

```

15. (*) Задача о рюкзаке

Даны n предметов и запросы «можно ли набрать вес w_i , используя только предметы с номерами от l_i до r_i ». При этом все $w_i \leq s$. Сделайте предподсчёт за $\mathcal{O}(ns \log s)$ так, чтобы на запрос можно было бы в online ответить за $\mathcal{O}(s \log s \log n)$.

Разбор задач практики

1. Рюкзак

$P(x) = \prod_i (1 + x^{a_i})$. Ответ – коэффициент при x^S .

2. Билетики

$P(x) = (x^0 + x^1 + \dots + x^9)^n$. Ответ = $\sum_s p_s^2$.

Подсчёт: бинарное возведение в степень, $Mul(n) + Mul(\frac{n}{2}) + \dots = \mathcal{O}(n \log n)$.

3. Возведение в степень

Возведём в степень за $\mathcal{O}(\log n)$, получаем $\text{FFT}(n) + \text{FFT}(\frac{n}{2}) + \text{FFT}(\frac{n}{4}) + \dots = \mathcal{O}(n \log n)$.

(*) Длина ответа $\frac{3}{10}n$, а в системе счисления 10^k даже меньше, $\frac{3}{10^k}n$. Берём $k = 5$, получаем при $n \leq 10^6$ типа `long double` хватает ($10^5 \cdot 10^5 \cdot \frac{3}{50}10^6 < 10^{15}$). 2 вещественных в одном комплексном $\Rightarrow$ умножение чисел = $2 \cdot \text{FFT}$, а время работы алгоритма $\leq 4 \cdot \text{FFT}$.

Итого $4 \cdot N \cdot \log N$ умножение комплексных чисел для $N = \frac{3}{50}10^6 < 2^{16}$.

$4 \cdot 2^{16} \cdot 16 = 4\,000\,000$ умножений комплексных чисел для задачи «посчитать 2^n при $n = 10^6$ ».

4. Рекуррентное соотношение

Возьмём $F(x) = \sum_{i=0..n} f_i x^i$, $A(x) = 1 - \sum_{i=1..k} a_i x^i$. Посчитаем $R(x) = F(x)A(x)$.

Посмотрим на коэффициенты $r_k, r_{k+1}, \dots, r_{k+n}$, должны быть нули.

5. Поиск с ошибками

Считаем отдельно для каждого символа s . Фиксируем символ s и строим

$$S_c = \sum_{s_i=c} x^i, T_c = \sum_{t|t-i+1=c} x^i \text{ (то есть } t \text{ разворачиваем)}$$

$P = S_c T_c$, тогда P_i равно числу позиций, где s есть и в s , и в приложении s к i -й позиции t .

6. Поиск с ошибками и шаблоном

То же самое, но в T_c учитываем не только s , но и «?».

В конце нужно вычесть пары «?», «?», все они учтены дважды.

7. Дуэль!

Если i -я клетка центр тройки (j, i, k) , то $j + k = 2i$.

Смотрим на многочлен $a = \sum a_i x^i$. $b = a^2$, b_{2i} равно числу нужных пар. Ответ = $\sum b_{2i} - 1$.

Можно обобщить на случай, когда в клетке растёт $a_i \in \mathbb{Z}$ деревьев, тогда ответ $\sum b_{2i} a_i - a_i^2$.

8. 3-SUM

Рассмотрим $A(x) = \sum_i x^{a_i}$, нам нужно посмотреть коэффициент при x^S у $A^3(x)$.

9. Уравнение

$a[x^n \bmod m]++$, для всех $x \in [0, m-1]$. Это делается за $\mathcal{O}(m \log n)$. И даже за $\mathcal{O}(\frac{m}{\log m} \log n)$:

$(xy)^n = x^n y^n \Rightarrow$ степень считать нужно только для простых. Далее $b = a^2, res = \sum b_i a_i$.

10. Цепочка умножений

Если сделаем k прямых $\text{FFT}(nk)$, перемножим, затем обратное $\text{FFT}(nk)$, то будет время $\mathcal{O}(k^2 n + \text{FFT}(nk)) = \mathcal{O}(nk(k + \log n))$. А можно перемножить пары соседних многочленов за $\text{FFT}(2n)$ каждую, потом новые пары соседних за $\text{FFT}(4n)$ каждую и т.д. Суммарно за $\mathcal{O}(\frac{k}{2} \text{FFT}(2n) + \frac{k}{4} \text{FFT}(4n) + \dots + \text{FFT}(kn)) = \mathcal{O}(nk \log(nk))$.

11. FFT по простому модулю

Ищем корни из единицы по модулю p . Для этого нужен первообразный корень g по модулю p . Теперь у нас есть циклическая группа $1 = g^0, g^1, g^2, \dots, g^{p-1} = 1$. Она аналогична по свойствам группе $w^0, w^1, \dots, w^{p-1} = w^0$ для $w = e^{2\pi i/(p-1)}$. Единственная проблема, что $p - 1 = 2^s t$ не обязательно степень двойки. С этим можно бороться двумя способами: взять группу размера 2^s , порождённую g^t , или объявить вершину рекурсии с многочленом длины t листом рекурсии и обработать его за $\mathcal{O}(t^2)$. Обычно берут $p = c2^k + 1$, где c мало. Время работы FFT для группы размера 2^s равно $\mathcal{O}(2^s s)$. Если степень многочлена-результата сильно меньше 2^s , мы можем взять группу любого меньшего размера 2^{s-j} .

12. (*) Поиск подстроки в строке

Посчитаем $errors_i = \sum (s_{i+j} - p_j)^2 = \sum s_{i+j}^2 + \sum p_j^2 - 2 \sum s_{i+j} p_j$, получили две частичные суммы квадратов и скалярное произведение циклических сдвигов. FFT!

13. (*) FFT и повышение точности

Заметим, что коэффициенты ответа при перемножении, как над $K[\mathbb{Z}]$, не более 10^{23} .

Решение #1. FFT по простому модулю и КТО. 3 раза перемножим многочлены по разным трём простым модулям порядка 10^9 (для этого нам достаточно типа `int64`). Теперь независимо для каждого из коэффициентов ответа воспользуемся КТО: есть остатки по трём модулям...

Решение #2. Представим коэффициенты многочлена A в виде $a_i = b_i + M c_i$, где $M = \lceil \sqrt{10^9} \rceil$, и $0 \leq b_i, c_i < M$. Теперь многочлен $A(x)$ представлен в виде $B(x) + C(x)M$. У нас есть два многочлена, представим так оба, считаем $A_1(x)A_2(x) = (B_1(x) + C_1(x)M)(B_2(x) + C_2(x)M) = B_1(x)B_2(x) + B_1(x)C_2(x)M + B_2(x)C_1(x)M + B_2(x)C_2(x)M^2$. То есть, сделаем 4 FFT с малыми коэффициентами вместо 1 FFT с большими. Также заметим, что коэффициенты произведений не больше $M^2 10^6 = 10^9 10^6 = 10^{15}$, поэтому типа `long double` нам точно хватит.

14. (*) Inplace FFT

`rev[rev[i]] = i` (то есть `rev` — инволюция), так что все циклы в перестановке имеют длину не более 2.

```

1 def FFT(a):
2     for (int i = 0; i < N; i++)
3         if (i < rev[i])
4             swap(a[i], a[rev[i]]);
5     for (int k = 1; k < N; k *= 2)
6         for (int i = 0; i < N; i += 2 * k)
7             for (int j = 0; j < k; j++)
8                 num tmp = root[k + j] * a[i + j + k];
9                 a[i + j + k] = a[i + j] - tmp;
10                a[i + j] += tmp;
```

15. (*) Задача о рюкзаке

Есть рюкзаки A и B , объединим их в рюкзак C : $C[i] = \vee(A[j] \wedge B[i-j])$. Можно делать это с FFT за $s \log s$. После этого просто построим ДО на предметах, в вершине рюкзак на отрезке предметов. Запрос разбивается на $\log n$ рюкзаков, которые надо объединить.

Домашнее задание

3.1. Обязательная часть

1. (2) Циклические сдвиги

Даны A, B , $|A| = |B| = n$. Найти D – такой циклический сдвиг B , что скалярное произведение A и D максимально.

2. (2) AVL деревья

Дано $n < 2^{16} = N = 2^H$. И число $h \leq 16 = H$.

Найти количество AVL деревьев глубины h из n вершин по модулю $3 \cdot 2^{18} + 1$.

- а) (1) Используя вещественное FFT, за $\mathcal{O}(NH^2)$.
- б) (1) Используя FFT по простому модулю, за $\mathcal{O}(NH^2)$.
- в) (+1) $\mathcal{O}(NH)$.

3.2. Дополнительная часть

1. (2) Динамика по дереву

Сколько способов вырезать из полного бинарного дерева глубины k поддерево размера s , содержащее корень исходного? Где здесь FFT? (1) $k = 100, s = 10^3$; (1) $k = 100, s = 10^5$.

2. (2) Раскрась-ка!

С помощью FFT за $\mathcal{O}^*(2^n)$ проверьте, можно ли вершины неорграфа раскрасить в k цветов.