Математические основы алгоритмов, весна 2023
Задание 1

1) По данным аргументам $a_0 < a_1 < \ldots < a_{n-1}$, эффективно вычислить коэффициенты полинома степени n с нулями ровно в этих аргументах.

2) Докажите корректность метода быстрого вычисления циклической и косоциклической сверток, представленного в лекциях:

$$a \circledast_+ b = \tfrac{1}{n} F_{n,\omega^{-1}}((F_{n,\omega} a) \circ (F_{n,\omega} b))$$
$$a \circledast_- b = w_{n,\psi^{-1}} \circ \left(\tfrac{1}{n} F_{n,\omega^{-1}}((F_{n,\omega}(w_{n,\psi} \circ a) \circ (F_{n,\omega}(w_{n,\psi} \circ b)))\right)$$

где

$$a \circledast_\pm b = \sum_{0 \le j \le i} a_j b_{i-j} \pm \sum_{i<j<n} a_j b_{n+i-j} \quad 0 \le i < n$$
$$\psi \text{ — первообразный корень из единицы степени } 2n, \psi^2 = \omega$$
$$w_{n,\psi} = (1, \psi, \ldots, \psi^{n-1})^T$$

3) Схему алгоритма Карацубы для умножения полиномов можно представить в матричном виде. Пусть

$$a(x) = a'(x) + a''(x)x^m$$
$$b(x) = b'(x) + b''(x)x^m$$
$$c(x) = a(x) \cdot b(x) = c'(x) + c''(x)x^m + c'''(x)x^{2m}$$

Тогда

$$\begin{pmatrix} c'(x) \\ c''(x) \\ c''(x) \end{pmatrix} = P \cdot \left(\left(Q \cdot \begin{pmatrix} a'(x) \\ a''(x) \end{pmatrix} \right) \circ \left(Q \cdot \begin{pmatrix} b'(x) \\ b''(x) \end{pmatrix} \right) \right)$$

где

$$P = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \qquad Q = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}$$

(a) Опишите альтернативную рекурсивную схему (а именно, матрицу P), имеющую ту же трудоемкость, что и алгоритм Карацубы, где

$$Q = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & -1 \end{pmatrix}$$

(b) Объясните вид матрицы Q и соотношение между матрицами P и Q в исходной и альтернативной схемах алгоритма Карацубы.

(c) В алгоритме Карацубы каждый из перемножаемых полиномов из n членов разбивается на два полинома из $n/2$ членов, и между полученными меньшими полиномами выполняется три умножения. Опишите рекурсивную схему для умножения полиномов, где каждый из перемножаемых полиномов из n членов разбивается на три полинома из $n/3$ членов, и между полученными меньшими полиномами выполняется пять умножений. Дает ли эта схема выигрыш в трудоемкости по сравнению с алгоритмом Карацубы?

4) В анализе схемы алгоритма Карацубы для умножения двоичных чисел, в отличие от умножения полиномов, имеется одна тонкость, опущенная в лекции. Пусть

$$a = a' + a''2^m \qquad b = b' + b''2^m$$

где a, b — числа из n бит, a', a'', b', b'' — числа из $n/2$ бит. Одно из произведений, вычисляемых в ходе алгоритма, равняется $(a' + a'')(b' + b'')$. В отличие от умножения полиномов, неверно утверждать, что в каждом из сомножителей этого произведения $n/2$ бит: количество битов может быть $n/2 + 1$ за счет переноса при сложении. Объясните, как скорректировать анализ трудоемкости алгоритма, чтобы учесть этот факт.

5) Определим *нормированное дискретное преобразование Фурье (DFT)* вектора $a \in \mathbb{C}^n$, прямое и обратное, как $\frac{1}{\sqrt{n}} F_{n,\omega} \cdot a$ и $\frac{1}{\sqrt{n}} F_{n,\omega^{-1}} \cdot a$, соответственно.

(a) Докажите, что прямое и обратное DFT действительно взаимно обратны и в обычном (с множителями 1, $\frac{1}{n}$), и в нормированном (с множителями $\frac{1}{\sqrt{n}}$, $\frac{1}{\sqrt{n}}$) определении.

(b) Пусть $a, b \in \mathbb{C}^n$, вектор b — нормированное DFT вектора a. Докажите *теорему Парсеваля*: $\|a\| = \|b\|$, где $\|x\| = \left(\sum_{0 \le i < n} |x_i|^2\right)^{1/2}$.

(c) При каких k, k-кратное применение нормированного прямого DFT с фиксированными n, ω равносильно нормированному обратному DFT? Тождественному преобразованию?

6) Проиллюстрируйте вычисления алгоритмом Шенхаге-Штрассена произведения чисел 1011011_2 и 10001111_2: приведите параметры алгоритма (значения модулей, длину блока ℓ, параметры FFT) и промежуточные вычисления для верхнего уровня рекурсии.

7) Докажите, что сложность (в битовых операциях) $M(n)$ алгоритма Шенхаге-Штрассена перемножения чисел размера n действительно $O(n \log n \log \log n)$. (Из лекции известно, что $M(n) = O(n \log n + mM(2\ell) + n)$. Докажите, что $M'(x) = c' \log x \log \log x$, где $M'(x) = \frac{M(x)}{x}$, а c' — константа.)

8) **Поиск подстроки с помощью FFT.** Дан текст $t = [t_0, \ldots, t_{n-1}] \in \{-1, 1\}^n$ и образец $p = [p_0, \ldots, p_{m-1}] \in \{-1, 1\}^m$, где $m \le n$.

(a) Как с помощью алгоритма FFT найти все вхождения p в t за время $O(n \log m)$? (Под временем имеется в виду количество операций над целыми числами.)

(b) Обобщите получившийся алгоритм на алфавит размера $\sigma > 2$; требуемое время работы алгоритма $O(n \log m \log \sigma)$.

(c) Обобщите (b) на случай, когда образец может содержать вхождения дополнительного символа '$*$' ("джокер"), соответствующего любому символу в тексте.

(d) Улучшите время работы алгоритма без джокеров до $O(n \log m)$. (Подсказка: представьте символы текста и образца положительными целыми числами t_i, p_j и рассмотрите квадрат разности $(t_i - p_j)^2$.)

(e) Обобщите (d) на случай, когда образец может содержать джокеры.

(f) Обобщите (e) на случай, когда текст также может содержать джокеры.