1 Неделя 1

1.1 Письменная часть

1. Если у строки w есть периоды p и q, где $|w| >= p + q - \gcd(p, q)$, то $\gcd(p, q)$ также является периодом этой строки. Приведите контрпример для теоремы без условия $|w| >= p + q - \gcd(p, q)$.

2. Строки Фибоначчи. Определим $F_0 = \varepsilon$, $F_1 = b$, $F_2 = a$, $F_n = F_{n-1}F_{n-2}$. Докажите, что существует k такое, что для $n \geq k$ выполнено F_n^2 — префикс F_{n+2}.

3. Строки Фибоначчи. Докажите, что существует k такое, что если $n \geq k$, то строка $F_n[1...|F_n| - 2]$ — палиндром.

1.2 Устная часть

1. Определим строку Туе-Морса: $T_n = t_0t_1t_2...t_{2^n-1}$, где $t_i = 0$, если двоичная запись числа i содержит четное число единиц, и $t_i = 1$ в противном случае. Доказать, что не существует двух равных как строки подстрок строки T_n, имеющих пересекающиеся вхождения в T_n.

2. Посчитать количество строк длины n на алфавите размера m, не содержащих заданную строку s как подстроку за полиномиальное от n, m и $|s|$ время.

3. Посчитать количество строк длины n на алфавите размера m, не содержащих заданную строку s как подстроку за полиномиальное от размера входных данных время.

4. Дана строка s. Посчитать матрицу $A : ||a_{ij}|| = \mathrm{LCP}(s[i..n-1], s[j..n-1])$; $i, j \geq 0$ за $\mathcal{O}(|s|^2)$. (LCP — наибольший общий префикс двух строк). Предложите алгоритм, который вычисляет число различных подстрок на каждом префиксе строки s за $\mathcal{O}(|s|^2)$

5. Предложите алгоритм вычисления матрицы c_{ij} = число различных подстрок в $s[i..j]$ за $\mathcal{O}(|s|^2)$ с помощью матрицы A из предыдущего занятия.

6. Заданы n, m и массив $p[i]$, который является префикс-функцией какой-то строки. Предложите алгоритм вычисления числа строк длины n с префикс-функцией p из букв состоящих из алфавита мощности m за время $\mathcal{O}(n)$.

7. У вас есть массив строк длины n. Вы сделали какую-нибудь из сортировок, которая делает $\mathcal{O}(n \log n)$ сравнений. Докажите, что сортировка работает за $\mathcal{O}(S \log S)$, где $S = \sum |s_i|$.

8. У вас есть изначально пустое двоичное дерево, которое поддерживает добавление за $\mathcal{O}(\log n)$ в среднем. Вы в него сделали `insert` n строк. Докажите, что это работает за $\mathcal{O}(S \log S)$ времени.

2 Неделя 2

2.1 Устная часть

9. Вам заданы два массива чисел a и b. Найдите все подмассивы $a[i..i + |b| - 1]$ такие, что существует x, что $a[i] = b[1] + x, a[i + 1] = b[2] + x, \ldots a[i + |b| - 1] = b[|b|] + x$ за время $\mathcal{O}(|a| + |b|)$.

10. Задана строка. Пусть $p_1[i]$ — максимальная длина палиндрома нечетной длины с центром в позиции i. $p_0[i]$ — аналогично для четной длины. Модифицировать алгоритм поиска z-функции для построения p_0 и p_1 за линейное время.

11. Как найти строку длины m в строке длины n с использованием z-функции и $\mathcal{O}(m)$ дополнительной памяти?

12. Предложите алгоритм, который за $\mathcal{O}(|s|)$ проверяет, что строка s простая.

13. Предложите алгоритм, который по строке вычисляет массив q. $q[i]$ — такое максимальное число, что $s[1..q[i]] = s[i-q[i]+1..i] = s[i..i+q[i]-1]$. Время работы алгоритма $\mathcal{O}(|s| \log |s|)$.

14. *Тандемным повтором* называется строка $w = \alpha\alpha$ для некоторой строки α. Предложите алгоритм для нахождения всех подстрок в строке s, которые являются тандемными повторами за время $\mathcal{O}(|s| \log |s|)$.

15. Вам заданы строки s и t длины n. А также перестановка π, тоже длины n. Найдите, какое минимальное число раз нужно применить перестановку π к строке s, чтобы получить строку t. Время работы $\mathcal{O}(n)$.

3 Неделя 3

3.1 Устная часть

16. Докажите три утверждения:

    (a) Если взять любую строку t, то $|\{\mathrm{LCP}(s_i, t) | 1 \le i \le n\}| = \mathcal{O}\left(\sqrt{\sum |s_i|}\right)$, где $\mathrm{LCP}(s_i, t)$ — длина наибольшего общего префикса s_i и t.

    (b) У любой вершины в боре, в который добавили все слова s_i, $\mathcal{O}\left(\sqrt{\sum |s_i|}\right)$ вершин предков, которые являются терминальными.

    (c) У любой вершины в боре, в который добавили все слова s_i, $\mathcal{O}\left(\sqrt{\sum |s_i|}\right)$ вершин предков, у которых хотя бы два ребенка.

17. У вас есть строка из цифр длины n. Вы можете порезать строчку на k непустых подстрок. Максимизируйте сумму чисел, полученных из этих подстрок за линейное время. Предложите алгоритм, как получить это число в той же системе счисления, в которой эти числа записаны в строке.

18. Дано n строк. Найдите количество пар индексов $(i, j), 1 \le i, j \le n$, что $s_i s_j$ является палиндромом за время $\mathcal{O}(\sum |s_i|)$.

19. Возьмем все простые строки над алфавитом $\{a, b\}$ длины, являющейся делителем n, упорядочим их лексикографически и сконкатенируем. Будем рассматривать получившуюся конкатенацию как зацикленную строку. Докажите, что в такой циклической строке все подстроки длины n различны, и множество всех этих подстрок совпадает с множеством строк длины n.

20. Задано множество слов w_i. Построив z-функцию для множества строк, научитесь за $\mathcal{O}(|t| + \sum |w_i|)$ отвечать на вопрос: сколько раз каждое слово входит в текст как подстрока, в какой позиции было первое вхождение каждого из слов, в какой последнее?

21. Задано множество слов w_i. Построив z-функцию для множества строк, за время $\mathcal{O}(\sum |w_i|)$ научитесь за $\mathcal{O}(|t| \log |t|)$ отвечать на вопрос: сколько слов входит в текст как подстрока, когда было самое первое вхождение какого-нибудь из слов?

22. Докажите, что число различных подстрок-палиндромов в строке длины n не более n.

23. Докажите, что если строки s и t таковы, что $st = ts$, то найдется такая строка p, что $s = p^i$ и $t = p^j$ для некоторых i и j.

24. У вас есть листок бумаги в клеточку $1 \times n$. Клетка i закрашена в цвет a_i. Вы можете сделать сгиб между какими-то двумя клетками и некоторые клетки полностью наложатся на какие-то другие: эти клетки должны быть одного цвета. Подобных сгибов можно делать сколько угодно. Узнайте, какой минимальной длины листочек может получиться, за время $\mathcal{O}(n \log n)$?

4 Неделя 4

4.1 Устная часть

25. Найдите наибольшую общую подстроку двух строк с помощью хеширования за время $\mathcal{O}(n \log n)$, где n — размер входных данных.

26. Вам задан текст t и слово w. Для каждой позиции в строке t определите, существует ли перестановка букв алфавита, что если ее применить к слову, то в этой позиции будет вхождение этого слова за время $\mathcal{O}(|t| + |w|)$ с помощью хеширования.

27. Предыдущая задача, но с нулевой вероятностью ошибки.

28. Посчитать число различных подстрок палиндромов в строке длины n за линейное время $\mathcal{O}(n)$.

29. Предыдущая задача, но с нулевой вероятностью ошибки.

30. Задано множество слов w_i. Построив автомат Ахо-Корасик для множества строк научитесь за $\mathcal{O}(|t|+\sum |w_i|)$ отвечать на вопрос: сколько раз каждое слово входит в текст как подстрока, в какой позиции было первое вхождение каждого из слов, в какой последнее?

31. Задано множество слов w_i. Построив автомат Ахо-Корасик для множества строк за время $\mathcal{O}(\sum |w_i|)$ научитесь за $\mathcal{O}(|t| \log |t|)$ отвечать на вопрос: сколько слов входит в текст как подстрока, когда было самое первое вхождение какого-нибудь из слов?

32. Посчитать количество строк длины n на алфавите размера m, не содержащих никакую из заданных строк $s_1, s_2, \ldots, s_k$ как подстроку за полиномиальное от длины входных данных время.

33. Задан набор строк $s_1, s_2, \ldots, s_k$. Предложите алгоритм, который вычисляет число строк длины n над алфавитом размера m, которые покрыты словами из s за полиномиальное от размера входных данных время. Строка покрыта множеством слов, если для любой позиции строки, существует вхождение какого-нибудь слова из множества, что это вхождение содержит эту позицию строки.

34. Задан набор строк $s_1, s_2, \ldots, s_k$. Предложите алгоритм, который вычисляет k-ю в лексикографическом порядке строку, среди строк длины n над алфавитом размера m таких, что они содержат хотя бы одно из слов из s, за полиномиальное от n, m и размера входных данных время.

35. Предложите модификацию алгоритма Ахо-Корасик, которая позволяет по тексту t за время $\mathcal{O}(t)$ найти все состояния, соответствующие каждому префиксу t, но при этом хранит только суффиксные ссылки и переходы в боре.

36. Дано 2 бора A и B. Для всех вершин u в A найти самую глубокую вершину v в B, соответствующую суффиксу u (префикс-функция бора в боре). Время работы $\mathcal{O}(|A| + |B|)$.

37. Задано n шаблонов. Существует ли бесконечная вправо строка, которая не содержит ни одного шаблона как подстроки. Определите это за полиномиальное от размера входа время.

38. Задано n шаблонов. Существует ли бесконечная в обе стороны строка, которая не содержит ни одного шаблона как подстроки. Определите это за полиномиальное от размера входа время.

39. Задан набор строк $s_1, s_2, \ldots, s_k$ суммарной длины S. Предложите алгоритм, который один раз ответит на много запросов вида: $\text{get}(l, r, t) = \sum\limits_{i=l}^{r} f(t, s_i)$, где $f(t, w)$ — число вхождений слова w в текст t. Время работы: $\mathcal{O}(\sum |t_i| + S + n\sqrt{S})$.

40. Решите предыдущую задачу за $\mathcal{O}(S + (n + \sum |t_i|) \log S)$.

41. Задано две матрицы $A_{n \times n}$ и $B_{m \times m}$ $(m \le n)$, состоящие из букв. Найдите все вхождения матрицы B в матрицу A за время $\mathcal{O}(n^2 + m^2)$.

5 Неделя 5

5.1 Устная часть

42. Задано две матрицы $A_{n \times n}$ и $B_{m \times m}$ $(m \le n)$, состоящие из букв. Найдите все вхождения матрицы B в матрицу A за время $\mathcal{O}(n^2 + m^2)$ **без использования хеширования**.

43. Предложите алгоритм, имея суффиксный массив и LCP, посчитать число различных подстрок на каждом префиксе строки за $\mathcal{O}(n \log n)$. И бонус к этому заданию: как это сделать за $\mathcal{O}(n)$.

44. Рассмотрим алгоритм (Kasai, Arimura, Arikawa, Lee, Park) для суффиксного массива циклических сдвигов строки без конечного символа. Покажите контрпример, на котором он работает некорректно. Покажите, как найти LCP для соседних в лексикографическом порядке циклических сдвигов строки.

45. Задана строка и перестановка. Предложите алгоритм, который за $\mathcal{O}(n)$ проверяет, что перестановка является суффиксным массивом строки без использования хеширования.

46. Задана строка длины n и много шаблонов. Для каждого шаблона в онлайн найдите число вхождений в строку за время $\mathcal{O}(L \log n)$, где L — длина шаблона.

47. Сделайте это же за время $\mathcal{O}(L + \log n)$.

48. Задана строка. Для каждого суффикса $s[i..n]$ определите больше он лексикографически суффикса $s[i+1..n]$ или нет за время $\mathcal{O}(n)$.

49. Задан суффиксный массив, построенный по строке, состоящей из символов двоичного алфавита. Восстановите строку за $\mathcal{O}(n)$.

50. Задана строка длины n. Найдите самую длинную подстроку, которая встречается в строке хотя бы дважды, не пересекаясь. Время: построение суффиксного массива + $\mathcal{O}(n)$.

51. Задана строка длины n и число k, найдите саму длинную подстроку, которая имеют хотя бы k вхождений в строку за построение суффиксного массива + $\mathcal{O}(n)$.

52. Задана строка длины n и число k, посчитать число различных подстрок строки, которые имеют хотя бы k вхождений в строку за $\mathcal{O}(n \log n)$.

53. Заданы строки суммарной длины n. Предложите алгоритм, который с помощью хеширования за $\mathcal{O}(n \log^2 n)$ находит наидлиннейшую подстроку, входящую во все строки.

54. Заданы строки суммарной длины n. Предложите алгоритм, который с помощью построения суффиксных массивов суммарной длины n и $\mathcal{O}(n)$ дополнительного времени находит наидлиннейшую подстроку, входящую во все строки.

6 Неделя 6

7 Неделя 7

7.1 Контрольная работа

8 Неделя 8

8.1 Устная часть

55. Предложите алгоритм посчитать число различных подстрок в каждом префиксе строки с использованием суффиксного дерева.

56. Предложите линейный алгоритм построения суффиксного дерева по заданному суффиксному массиву и LCP.

57. Придумайте как по суффиксному массиву и LCP построить точки (x_i, y_i) за линейное время, что небинарное декартово дерево будет суффиксным деревом строки, по которой построен суффиксный массив и LCP. *Небинарное декартово дерево* — такое же дерево поиска как и декартово, но если в одно поддереве у вершин одинаковые приоритеты, то они сливаются в одну вершину, тем самым у нее может быть больше двух детей. Предложите линейный алгоритм построения такого дерева.

58. Решите с помощью суффиксного дерева за линейное время: Задана строка длины n. Найдите самую длинную подстроку, которая встречается в строке хотя бы дважды, не пересекаясь.

59. Решите с помощью суффиксного дерева за линейное время: Задана строка длины n и число k, найдите саму длинную подстроку, которая имеют хотя бы k вхождений в строку.

60. Решите с помощью суффиксного дерева за линейное время: Заданы строки суммарной длины n. Предложите алгоритм, который находит наидлиннейшую подстроку, входящую во все строки.

61. Решите с помощью суффиксного дерева за линейное время: Задана строка длины n и число k, посчитать число различных подстрок строки, которые имеют хотя бы k вхождений в строку.

62. Задана строка s длины n. Предложите алгоритм, отвечающий на запрос в онлайн: «Заданы i и j: Сколько раз $s[i..j]$ входит в s как подстрока?». Время работы $\mathcal{O}(\log n)$ на запрос и $\mathcal{O}(n \log n)$ предпосчета.

63. Задано n строк s_i. Для каждой строки найдите ее минимальный префикс, который не является префиксом никакой другой строки. Решите за линейное от размера входных данных время.

64. Задана строка s длины n и m запросов $\langle L, R\rangle$. Ответьте на запросы в online за время $\mathcal{O}(\log n)$: максимальная подстрока палиндром в $s[L..R]$.

65. Задана строка s. Ответьте на запросы в online: задано k подстрок, найдите две из них, у которых максимальный общий префикс за время $\mathcal{O}(k \log{(n + k)})$ на запрос.

66. Решите с помощью суффиксного дерева за линейное время: Пусть $Q(s)$ — множество всех подстрок строки s. Заданы строки s и t. Посчитайте размер множества $\{xy | x \in Q(s) \land y \in Q(t)\}$.

67. Задано два текста из букв и пробелов. Оба текста не начинаются и не заканчиваются на пробелы, а также не содержат два пробела подряд. Разрешается в первом тексте буквы на некоторых позициях поменять на пробелы, а потом заменить подряд идущие пробелы на

один пробел, удалить пробелы с начала и с конца текста. Какие позиции нужно заменить на пробелы, чтобы тексты стали одинаковыми. Решите за линейное время.

68. Решите с помощью суффиксного дерева за линейное время: Для каждой строки найдите ее минимальную подстроку, которая не является подстрокой никакой другой строки.

9 Неделя 10

9.1 Письменная часть

1. Докажите, что наибольший общий делитель последовательных чисел Фибоначчи равен 1.

2. Докажите, что x — общий делитель a и b ($a \le b$) тогда и только тогда, когда x — общий делитель a и $b - a$.

3. Докажите, что функция $\sigma_k(n) = \sum\limits_{d|n} d^k$ — мультипликативная.

4. Пусть $n = pq$, где p и q — различные простые числа. Покажите, как, зная n и $\varphi(n)$, найти p и q.

9.2 Устная часть

69. Найдите сумму всех чисел от 0 до $n-1$, взаимнопростых с n.

70. Выведите формулу для $\varphi(n)$, используя формулу включений-исключений.

71. Посчитайте $\binom{n}{k} \bmod p$ за $O(p \log n)$ (примечание: найдите $\frac{n!}{p^k} \bmod p$, где k — максимальная степень вхождения простого p в $n!$).

72. Пусть $A_n = x^n - 1$, где x — какое-то натуральное число. Докажите, что $\gcd(A_n, A_m) = A_{\gcd(n,m)}$.

73. Пусть F_n — n-е число Фибоначчи ($F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}$. Докажите, что $\gcd(F_n, F_m) = F_{\gcd(n,m)}$.

74. Каков критерий существования решения и алгоритм восстановления числа в КТО, если убрать требование взаимной простоты модулей m_1 и m_2?

75. Заданы числа m, x, l, r ($0 \le l \le r < m$). Найдите минимальное k, что $(k \cdot x) \bmod m \in [l, r]$ за $\mathcal{O}(\log m)$.

76. Посчитайте количество пар (x, y), что $1 \le x \le n$, $1 \le y \le m$, и $\frac{y}{x} \le \frac{a}{b}$ за $O(\log N)$, где N — максимальное число среди заданных.

77. Покажите, как найти обратные по модулю m к числам $1, 2, \ldots, n$ за $\mathcal{O}(n)$ алгебраических операций.

78. Предложите модификацию расширенного алгоритма Евклида, которая использует только операции сложения, вычитания, сравнения чисел и умножения/деления на степень двойки. Время работы — $\mathcal{O}(\log n)$ алгебраических операций.

79. Для заданных a, b и c найдите количество решений диофантова уравнения $ax + by = c$ с $x, y \ge 0$ за $O(\log \min(a, b))$.

10 Неделя 11

10.1 Письменная часть

1. Дзета-функция Римана: $\zeta(s) = \sum\limits_{n=1}^{\infty} n^{-s}$. Покажите, что $\zeta(s) = \prod\limits_{p \in P} (1 - \frac{1}{p^s})^{-1}$.

2. Докажите, что $\sum\limits_{d|n} \varphi(d) = n$

3. В схеме RSA опасно брать простые, близкие друг к другу. Покажите, как факторизовать $n = pq$ за $\mathcal{O}(|p - q| \cdot \text{poly}(\log n))$.

11 Неделя 11

11.1 Устная часть

80. Используя $\ln(1 + x) \le x$ и $\sum\limits_{n=1}^{N} \frac{1}{n} = \ln N + \mathcal{O}(1)$, докажите, что $\sum\limits_{p \in P, p \le N} \frac{1}{p} = \Theta(\ln \ln N)$.

81. Докажите, что $\varphi(n) = \Omega(n^{\frac{1}{2}})$.

82. Докажите, что $\varphi(n) = \Omega(\frac{n}{\log n})$.

83. Докажите, что $\varphi(n) = \Omega(\frac{n}{\log \log n})$.

84. Решите задачу дискретного логарифма для простого модуля p вида $2^k + 1$ за $\mathcal{O}(\text{poly}(k))$.

85. Предполагая, что мы знаем факторизацию $p-1$, обобщите решение предыдущей задачи для любого простого модуля p за $\mathcal{O}(\sqrt{r}\,\text{poly}(\log p))$, где r — максимальный простой делитель $p - 1$.

86. Зафиксируем простое p и остатки g и h. Рассмотрим хеш-функцию $h(x, y) = g^x h^y$. Как, найдя коллизию в такой хеш-функции, решить задачу дискретного логарифма? Предложите альтернативное решение задачи дискретного логарифма за $\mathcal{O}(\sqrt{p})$ на основе этого факта.

87. Для простого p и заданных $a \in [1..p - 1]$ и k найдите все решения $x^k \equiv a \pmod{p}$ за $\mathcal{O}(\sqrt{p})$.

88. Тест Ферма на простоту. Рассмотрим следующий алгоритм проверки числа n на простоту: берем случайное $a \in [1 \ldots n)$. Если $\gcd(a, n) \ne 1$, выдать, что n — составное. Если $a^n \ne a$, выдать, что n — составное. Повторить несколько раз, если для всех выбранных a тест пройден, сказать, что число простое. Этот тест неверен, такие n, что $\forall a \in [1 \ldots n) a^n = a$, существуют и называются **числами Кармайкла**. Докажите, что если n — число Кармайкла, то $n = p_1 \cdot p_2 \cdot \ldots \cdot p_k$ для некоторого набора простых $\{p_i\}$ и $p_i - 1 \mid n - 1$ для всех p_i.

89. Докажите, что если $n = p_1 \cdot p_2 \cdot \ldots \cdot p_k$ для некоторого набора простых $\{p_i\}$ и $p_i - 1 \mid n - 1$ для всех p_i, то n — число Кармайкла.

90. Докажите, что любое число Кармайкла n нечетно, имеет как минимум три простых делителя и все простые делители $p < \sqrt{n}$.

91. Атака на RSA. Предположим, что Алиса отправила одно и то же сообщение m, используя один и тот же модуль $n = pq$, но разные публичные экспоненты e_A и e_B, причем $\gcd(e_A, e_B) = 1$. Покажите, как Еве, зная $m^{e_A} \bmod n$ и $m^{e_B} \bmod n$, восстановить m.

92. В RSA часто используется публичная экспонента e небольшого размера и с небольшим числом единичных битов 3 или $65537 = 2^{16}+1$. Как это помогает ускорить шифрование? При $e = 3$ посылка одного сообщения трем разным адресатам (по разным модулям) приводит к возможности расшифровки. Как?

12 Неделя 12

12.1 Устная часть

93. Обобщите алгоритм Тонелли-Шенкса для решения уравнения $x^3 = a \pmod p$ за $\mathcal{O}(\text{poly}(\log p))$.

94. Решите уравнение $x^q = a \pmod p$ за $\mathcal{O}(q \cdot \text{poly}(\log p))$ для простого q.

95. Решите уравнение $x^q = a \pmod p$ за $\mathcal{O}(\sqrt{q} \cdot \text{poly}(\log p))$ для простого q.

96. Решите задачу дискретного логарифма (нахождение x: $a^x = b \pmod n$) за $O(\sqrt{x})$

97. Докажите, что n – простое тогда и только тогда, когда $\binom{n}{k} \equiv 0 \pmod n$ для всех $0 < k < n$.

98. Заданы числа n и a; $(n, a) = 1$. Докажите, что n — простое тогда и только тогда, когда $(x + a)^n \equiv x^n + a \pmod n$ (как многочлены).

99. Покажите, что если n — составное и не является числом Кармайкла, то тест Ферма корректно определяет, что число составное, с вероятностью $\geq \frac{1}{2}$.

100. Тест Лукаса на простоту: пусть известна факторизация $n - 1$. Докажите, что n простое $\iff$ существует взаимнопростое a, что $a^{n-1} = 1 \pmod n$, $a^{\frac{n-1}{q}} \neq 1 \pmod n$, где q — любой простой делитель $n - 1$. Составьте на основе этого утверждения алгоритм проверки n на простоту.