

# Вопросы к зачёту для группы А'

## ЛКШ.2017.Июль, «Электроник»

### Структуры данных

1. (3) RMQ: Формулировка. Простые решение:  $[O(n^2), O(1)]$ ,  $[O(1), O(n)]$ ,  $[O(n), O(\log n)]$ .
2. (4) RMQ: Sparse table:  $[n \log n, 1]$ ,  $[n, \log \log n]$ ,  $[n \log^* n, \log^* n]$ .
3. (4) RMQ $\pm 1$  за  $[n, 1]$ .
4. (3) LCA. Решение за  $O(\text{distance})$ . Двоичные подъёмы  $[n \log n, \log n]$ .
5. (4) LCA  $\rightarrow$  RMQ $\pm 1$  за  $O(n)$ , решение LCA за  $[O(n), O(1)]$  (Эйлеров обход). Функция на поддереве.
6. (4) RMQ  $\rightarrow$  LCA за  $O(n)$  (построение декартова дерева за  $O(n^2)$  и  $O(n)$ ), итого RMQ за  $[O(n), O(1)]$ .
7. (4) LCA в Offline за  $O((n+m)\alpha)$  (алгоритм Ахо-Ульмана-Тарьяна).
8. (3) Функция на пути в статичном дереве. Произвольная. Обратимая.
9. (4) Сумма на пути в дереве с меняющимися весами на рёбрах.
10. (3) LA: решение в offline и решение двоичными подъёмами.
11. (3) Дерево отрезков с операциями снизу: память, время, сравнение с версией сверху.
12. (3) Динамическое дерево отрезков (и снизу, и сверху).
13. (4) Персистентное дерево отрезков с групповыми операциями.
14. (4) Дерево отрезков сортированных массивов, 2D запросы на плоскости и на массиве. Дерево отрезков декартовых деревьев. Пример задачи.
15. (4) Площадь объединения прямоугольников.
16. (4) Количество точек в прямоугольнике в online с персистентной сканирующей прямой.
17. (5) Поиск пересекающихся отрезков. Поиск объединения невыпуклых многоугольников.
18. (4) Задача: для каждого из направлений из  $(0, 0)$  найти ближайший из  $n$  непересекающихся отрезков.
19. (3) Корневая на массиве: сумма (два способа), минимум.
20. (3) Фаброзавры: найти количество чисел равных  $x$  на отрезке.
21. (4) Корневая со split/rebuild, корневая со split/merge. Запросы: add, del, get(l, r, x), reverse(l, r).
22. (4) Отложенные операции (корневая по запросам). Пример: динамический массив.
23. (4)  $k$ -я порядковая на отрезке за  $O(\log^3 n)$  и  $O(\log^2 n)$  бинпоиском по ответу.
24. (3) Задача: дан массив из нулей и единиц, найти  $k$ -ую единицу (решения за  $\log^2 n$  и  $\log n$ ).
25. (5)  $k$ -я порядковая на отрезке за  $O(\log n)$  параллельным спуском по двум версиям.
26. (4) DCP. Решение корневой за  $O(m^{3/2})$ .
27. (5) DCP. Решение деревом отрезков по запросам без СНМ.

### Строки

28. (3) Сравнение подстрок строки. Поиск подстроки в строке с  $O(1)$ .
29. (3) Выбор пары  $(P, M)$ , вероятность ошибки для задач “поиск подстроки”, “число различных подстрок”.
30. (4) Хеш-таблица списками. Хеш-таблица открытой адресацией (с удалением).
31. (4) Поиск общей подстроки двух и  $k$  строк за  $O(n \log n)$
32. (3) Несжатый бор. Хранение рёбер (3 способа). Поиск словарных слов в тексте.
33. (4) Ахо-Корасик: построение полного автомата. Поиск словарных слов в тексте.
34. (4) Ахо-Корасик: обобщение префикс функции, ленивая функция построения. Поиск слов в тексте.
35. (3) map: string  $\rightarrow$  int – два способа.
36. (4) Дан массив строк, запрос: “количество строк, начинающихся на s, кончающихся на t”.
37. (4) Даны текст и словарь: для каждого слова словаря посчитать, сколько раз оно встречается в тексте.
38. (3) Суффиксный массив: определение, хранение в памяти:  $O(n)$ . Построение за  $O(n^2)$ .

39. (3) Сравнение строк хешами на больше-меньше и построение суффиксного массива за  $\mathcal{O}(n \log^2 n)$ .
40. (4) Построение суффиксного массива за  $\mathcal{O}(n \log n)$ . Суффиксы  $\Leftrightarrow$  циклические сдвиги.
41. (3) LCP: полный подсчет за  $\mathcal{O}(n^2)$ , вычисление хешами LCP(i,j) за  $\mathcal{O}(\log n)$ .  
Вычисление LCP всех соседних за  $\mathcal{O}(n)$ . Вычисление LCP произвольных за  $\mathcal{O}(1)$ .
42. (5) Поиск подстроки в тексте. Ответ на запрос за  $\mathcal{O}(|s| \log |T|)$ ,  $\mathcal{O}(|s| + \log |s| \log |T|)$ ,  $\mathcal{O}(|s| + \log |T|)$
43. (4) Задачи: количество вхождений строки, количество различных подстрок, общая подстрока  $k$  строк.
44. (3) Сжатый бор, суфф дерево. Хранение, построение за квадрат. Поиск подстроки в тексте.
45. (3) Число различных подстрок, max рефрен, число вхождений, первое и последнее вхождения.
46. (4) Общая подстрока двух строк,  $k$  строк, LZSS.
47. (3) Дерево  $\rightarrow$  массив + LCP за  $\mathcal{O}(n)$ .
48. (3) Массив + LCP  $\rightarrow$  дерево за  $\mathcal{O}(n)$ .
49. (4) Укконен: версия за квадрат, саморастущие листья, разветвится самый длинный суффикс.
50. (5) Укконен: вычисление и использование суффссылок, линейность времени работы.

### Паросочетания и потоки

51. (3) Паросочетание (M), вершинное покрытие (VC), независимое множество (IS). Биекция  $IS \leftrightarrow VC$ .
52. (3)  $\min |VC| \geq \max |M|$ , жадное приближение  $\max |M|$ .
53. (4) Лемма о дополняющем пути. Алгоритм поиска паросочетания, реализация.
54. (4) Алгоритм Куна: доказательство, оптимизации вида “не обнуляем пометки”.
55. (4) Построение VC и IS по паросочетанию за  $\mathcal{O}(E)$ .
56. (4) Паросочетание, которое максимизирует суммарный вес вершин первой доли. С доказательством.
57. (4) Рандомизированный алгоритм поиска паросочетания в произвольном графе.
58. (4) Задачи: доминошки на доске с дырками, покрытие ациклического графа путями.
59. (4) Задачи: максимальная двудольная клика, покрытие грида минимальным числом отрезков.
60. (5) Теорема Дилворта: поиск максимальной антицепи.
61. (4) Стабильное паросочетание. Алгоритм “мальчики предлагают, девочки отказываются”.
62. (3) Поток. Разрез. Величина потока и разреза. Циркуляция. Декомпозиция. Связь  $k$  путей и потока.
63. (4) Теорема и алгоритм Форда Фалкерсона поиска потока за  $\mathcal{O}(|f|E)$
64. (4) Декомпозиция потока за  $\mathcal{O}(E^2)$ . За  $\mathcal{O}(VE)$ .
65. (3) Хранение графа: вектора, хранение соседних рядом  $(2i, 2i+1)$ .
66. (3) Вершинно непересекающие пути. Хранение неорграфа. Нахождение паросочетания. Несколько истоков и стоков.
67. (4) Алгоритм Эдмондса-Карпа. Лемма:  $d[s, v]$  растут.
68. (4) Алгоритм масштабирования потока.
69. (4) Задачи: выделение  $k$ -регулярного подграфа двудольного, восстановление матрицы по суммам строк и столбцов, восстановление турнирной таблицы.
70. (3) Определение стоимости, построение графа, представление неорграфа. Формулировка 4 задач.
71. (3) Критерий оптимальности: отсутствие отрицательных циклов.
72. (4) Алгоритм Клейна для поиска минимальной циркуляции и решение  $|f| = k$ .
73. (3) Поиск отрицательных циклов и цикла минимального среднего веса.
74. (4) Поиск потока величины  $k$  за  $\mathcal{O}(kVE)$ .
75. (4) Поиск потока величины  $k$  за  $\mathcal{O}(kV^2)$ .
76. (5) Алгоритм capacity scaling за  $\mathcal{O}(E \log U \cdot Dijkstra)$ .
77. (4) Диниц за  $\mathcal{O}(V^2E)$ .
78. (4) Диниц с масштабированием за  $\mathcal{O}(VE \log U)$ .
79. (5) Хопкрофт-Карп за  $\mathcal{O}(EV^{1/2})$

## Игры, геометрия, перебор

80. (3) Симметричные и несимметричные игры на ациклическом графе. Пример игры.
81. (4) Симметричные и несимметричные игры на графах с циклами. Ретроанализ.
82. (4) Длина игры в случае, когда победитель минимизирует/максимизирует длину игры.
83. (4) Ним. Проигрышная позиция. Доказательство ксору.
84. (3) Определение функции Гранди. Вычисление за  $\mathcal{O}(E)$ .
85. (4) Доказательство того, что функция Гранди от суммы игр равна ксору.
86. (4) Задачи: жестокая задача, Nucleon Bush, Ним с неубывающими размерами кучек.
87. (3) Выпуклая оболочка: заворачивание подарка за  $\mathcal{O}(nk)$ , Грехем и Эндрю за  $\mathcal{O}(n \log n)$ .
88. (4) Выпуклая оболочка: оценка снизу на время построения, алгоритм quick hull.
89. (4) Две самые дальние точки среди  $n$  данных.
90. (4) Самый большой треугольник, построенный на  $n$  данных точках.
91. (4) Узнать для прямой, пересекает ли она многоугольник: offline и online.
92. (5) Общая касательная к двум выпуклым непересекающимся многоугольникам.
93. (3) Проверка, внутри ли точка?
94. (4) Самая дальняя точка по направлению  $(A_i, B_i)$  среди данных  $n$  точек (опорная прямая).
95. (4) Пересечение многоугольника и прямой.
96. (5) Касательная к многоугольнику. Расстояние от точки до многоугольника.
97. (3) Макс клика перебором с запоминанием за  $\mathcal{O}(2^{n/2})$ .
98. (4) Макс клика расщеплением по вершине макс степени. Отдельная обработка вершин степени  $0, 1, n$ .
99. (4) Оптимизации: отсечение по времени, отсечение по ответу, iterative deepening.
100. (4) Гамильтонов путь: жадность, iterative deepening по  $k$ .
101. (4) Random walking, пример с 3-SAT.
102. (3) Meet-In-The-Middle, пример с сокобаном.
103. (5)  $A^*$  и использование в переборе.
104. (3) Антисимметричная игра на большом дереве.