

# Вопросы к экзамену по алгоритмам SPb HSE, 1 курс, июнь 2022

Собрано 28 августа 2022 г. в 12:15

## Общая информация

- Кроме конспектов полезно смотреть **разборы** задач из практик.
- *Курсивом* помечено то, что было разобрано на практике.
- (a) темы на 3 (оценка 4-5).
- (b) темы на 4 (оценка 6-7).
- (c) темы на 5 (оценка 8-9).
- (+) факультативные темы (оценка 10) – для оценки 10 нужно уметь рассказывать *четыре* такие темы (всего их шесть).

## Жадности и приближённые алгоритмы

- (a) 1. Алгоритм Хаффмана, доказательство, реализация кучей за  $\mathcal{O}(k \log k)$ .
- (b) 2. Построение кодов по массиву частот за  $\mathcal{O}(\text{sort}(k) + k)$ .
- (a) 3. Жадность. Задачи на сортировку. Способы доказательства с примерами задач: через swap соседних и через выбор первого/последнего элемента.
- (a) 4. Жадность. Задачи на сортировку. Примеры: выполнить побольше задач с общим дедлайном, файлы на ленте, *непрерывный рюкзак, башня из спортсменов*.
- (b) 5. Жадность. Задания с дедлайнами: выполнить все за  $\mathcal{O}(n \log n)$ , выполнить максимум за  $\mathcal{O}(n^2)$ .
- (c) 6. Жадность. Задания с дедлайнами: выполнить максимум. Решение  $\mathcal{O}(n \log n)$ . Доказательство корректности.
- (b) 7. Жадность. Задача про два (упорядоченных) станка. Формулировка. Корректное решение без доказательства.
- (b) 8. Правило Варнсдорфа для частного случая. Применение к общему случаю.
- (a) 9. TSP. 2-ОПТ решение.
- (b) 10. TSP. 1.5-ОПТ решение. *Неприближаемость без неравенства треугольника*.
- (a) 11. Рюкзаки. Формулировки: partition, knapsack, bin packing.
- (a) 12. Рюкзаки. В предположении NP-трудности partition показать трудность knapsack и bin packing.
- (a) 13. Partition. Жадное решение (без доказательства оценки).
- (a) 14. Partition. Алгоритм Кармаркар-Карпа (без доказательства оценок).
- (b) 15. Partition. PTAS-схема.
- (c) 16. Partition. Оценка жадности  $\frac{6}{5}$ ОПТ, обоснование  $\mathcal{O}(n^{-\log n})$  для Кармаркар-Карпа.
- (b) 17. Knapsack. *Жадность по удельной стоимости не даёт приближения. 2-приближение*.
- (b) 18. Knapsack. PTAS-схема.
- (c) 19. Knapsack. FPTAS-схема.
- (a) 20. Bin Packing. *Доказательство того, что First Fit, Next Fit – это 2-приближения*.
- (a) 21. Set Cover.  $\ln(n)$ -приближение. Решение без доказательства.
- (b) 22. Set Cover.  $\ln(n)$ -приближение. Доказательство оценки.
- (c) 23. Set Cover.  $\ln(n)$ -приближение во взвешенном случае. Реализация за линейное время в невзвешенном случае. 2-ОПТ приближение для Vertex Cover.

## Центроиды

- (a) 24. Центроидная декомпозиция: построение за  $\mathcal{O}(n \log n)$  времени,  $\mathcal{O}(n)$  памяти.
- (b) 25. Центроидная декомпозиция: подробности хранения, LCA, поиск минимума на пути за  $\mathcal{O}(\log n)$  и  $\mathcal{O}(\log \log n)$ .
- (c) 26. Центроидная декомпозиция: количество путей длины  $d$ .
- (c) 27. Центроидная декомпозиция: покраска вершин на расстоянии  $\leq d$ .

## Бинарные деревья поиска

- (a) 28. BST. Add, Del, Find, LowerBound за  $\mathcal{O}(h)$ , симметричный обход. Нижняя оценка на Add.
- (a) 29. BST. Next, Prev, Find за  $\mathcal{O}(1)$ . Итератор в `std::set`.
- (b) 30. BST. Обработка равных ключей. Удаление за  $\mathcal{O}(1)$ .
- (b) 31. BST. Персистентная версия добавления, удаления.
- (a) 32. AVL. Инвариант, оценка глубины. Add, перебалансировка.
- (b) 33. AVL. Del, Merge (без док-ва перебалансировки на  $k$ ). Персистентное вращение.
- (c) 34. AVL. Split (с док-вом перебалансировки на  $k$ ).
- (a) 35. Общие идеи: BST и неявный ключ, BST и персистентность.
- (b) 36. Общие идеи: запрос на отрезке BST-дерева, отложенные операции, разворот отрезка.
- (a) 37. Treap. Два определения случайного дерева (RBST). Эквивалентность определений. Treap является RBST.
- (a) 38. Treap. Матожидание глубины RBST (без доказательства) и глубины любого элемента RBST (с доказательством).
- (a) 39. Treap. Операции Split, Merge. Оценка времени работы. Неявный ключ.
- (b) 40. Treap. Эффективная реализация Add, Delete. Персистентные декартовы деревья.

## Структуры данных

- (a) 41. Persistent. Массив – любой способ.
- (a) 42. Persistent. Offline, дерево версий. Детская персистентность для массива (2 версии), частичная персистентность за  $\mathcal{O}(\log n)$ .
- (a) 43. Persistent. Стек.
- (b) 44. Persistent. СММ. Персистентный массив: BST и дерево отрезков.
- (c) 45. Persistent. Очередь.
- (b) 46. Persistent. Персистентный дек (Pairing Deque).
- (+) 47. Персистентность через Fat Nodes.
- (a) 48. Дерево отрезков. Реализация сверху. Оценка на память, на время get.
- (a) 49. Дерево отрезков. Массовые модификации. += и getMin. = и getSum.
- (b) 50. Дерево отрезков. Реализация снизу. Сравнение с реализацией сверху.
- (b) 51. Дерево отрезков. Динамическое (два способа), сжатие координат. Применение сжатия координат для задач со scanline.
- (a) 52. Scanline. Offline: число точек в прямоугольниках за  $\mathcal{O}(\log n)$ ; число прямоугольников, покрывающих точку за  $\mathcal{O}(\log n)$ .
- (b) 53. Scanline. Online: число точек в прямоугольниках за  $\mathcal{O}(\log n)$ ; число прямоугольников, покрывающих точку за  $\mathcal{O}(\log n)$ .
- (b) 54. Scanline. Площадь объединения прямоугольников.
- (a) 55. 2D. ДО из сортированных массивов. Связь с двумерным запросом (количеством точек в прямоугольнике за  $\mathcal{O}(\log^2 n)$ ).

- (b) 56. 2D. Поиск  $k$ -й порядковой статистики на отрезке за  $\mathcal{O}(\log^2 n)$ .
- (c) 57. 2D. Многомерные структуры. ДО из ДО (зачем нужно?), ДО из ДД (зачем нужно?). Трёхмерное ДО,  $k$ -мерное ДО.
- (c) 58. 2D. Поиск  $k$ -й порядковой на отрезке за  $\mathcal{O}(\log n)$ .
- (+) 59. Техника Fractional Cascading для двумерных запросов и параллельного бинпоиска.
- (+) 60. КД-дерево. Ответы на двумерные запросы (и двумерные модификации!) за  $\mathcal{O}(\sqrt{n})$ , запрос «перечислить все точки в прямоугольнике за  $\mathcal{O}(k + \log n)$ ».

## RMQ, LCA, LA

- (a) 61. RMQ. Разреженная таблица (Sparse Table).
- (b) 62. RMQ. Ускорения разреженной таблицы.  $\langle \mathcal{O}(n \log \log n), \mathcal{O}(1) \rangle$ ,  $\langle \mathcal{O}(n), \mathcal{O}(\log \log n) \rangle$ ,  $\langle \mathcal{O}(n \log^* n), \mathcal{O}(\log^* n) \rangle$ .
- (b) 63. RMQ. *Disjoint Sparse Table*. Ответ на запрос за  $\mathcal{O}(1)$ .
- (a) 64. RMQ. Эйлеров обход дерева. Запросы на поддеревьях.
- (a) 65. RMQ. Сведение LCA  $\rightarrow$  RMQ $\pm 1$ . Решение задачи LCA за  $\langle \mathcal{O}(n \log n), \mathcal{O}(1) \rangle$ .
- (b) 66. RMQ. Сведение RMQ  $\rightarrow$  LCA. Построение *Cartesian Tree* за линейное время.
- (c) 67. RMQ. Фарах-Колтон – Бендер. Метод четырёх русских, решение RMQ и LCA за  $\langle \mathcal{O}(n), \mathcal{O}(1) \rangle$ .
- (a) 68. LCA. Двоичные подъёмы. Решение LCA, два способа (с одной двигающейся вершиной и с двумя).
- (+) 69. LCA в offline, алгоритм Тарьяна.
- (a) 70. Сумма на пути в дереве за  $\langle \mathcal{O}(n), \mathcal{O}(1) \rangle$ .
- (c) 71. Сумма на пути в меняющемся дереве за  $\langle \mathcal{O}(n), \mathcal{O}(\log n) \rangle$ .
- (a) 72. LA. Решение LA в online за  $\langle \mathcal{O}(n \log n), \mathcal{O}(\log n) \rangle$  и в offline за  $\langle \mathcal{O}(n), \mathcal{O}(1) \rangle$ .
- (c) 73. LA. Алгоритм Вишкина – решение LA за  $\langle \mathcal{O}(n), \mathcal{O}(\log n) \rangle$ .
- (+) 74. LA: Вишкин за  $\langle \mathcal{O}(n \log n), \mathcal{O}(1) \rangle$ , решение LA через ladders decomposition.
- (+) 75. LA: решение за  $\langle \mathcal{O}(n), \mathcal{O}(1) \rangle$  методом четырёх русских.