

# Вопросы к экзамену по алгоритмам SPb HSE, 2-й курс, октябрь 2022

## 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) – нужно заботать 5 любых темы из 9.

## Деревья, структуры данных

- (b) 1. Rope. Skip-List. Операции Find, Insert.
- (c) 2. Rope. Skip-List. Операции Erase, Split, Merge.
- (a) 3. Splay. Три вида поворотов. Операции Splay, Add.
- (a) 4. Splay. Формулировка главной теоремы о времени работы.
- (b) 5. Splay. Del, *Split*, *Merge*.
- (b) 6. Splay. Потенциал. Оценка всего, кроме операции Splay.
- (c) 7. Splay. Доказательство амортизированной оценки операции Splay.
- (a) 8. Rope. Корневая. Через Split/Merge.
- (b) 9. Rope. Через Корневая. Split/Rebuild. Примеры.
- (b) 10. Rope. Корневая. Оптимальный выбор размера куска на примере задачи (`kth_stat + reverse`).
- (c) 11. *Корневая декомпозиция по запросам на примере «Dynamic Connectivity Offline за  $\mathcal{O}(m\sqrt{m})$ ».*
- (a) 12. LCA-Offline.
- (b) 13. RMQ-Offline.
- (c) 14. RMQ-Online на путях в дереве за  $\langle \mathcal{O}(\text{sort}(n)), \mathcal{O}(1) \rangle$ .
- (a) 15. Heavy-Light Decomposition. Определение, Построение за  $\mathcal{O}(n)$ .
- (a) 16. Heavy-Light Decomposition. Подробности реализации.
- (a) 17. Heavy-Light Decomposition. Функции на пути за  $\mathcal{O}(\log^2 n)$ .
- (a) 18. Heavy-Light Decomposition. Функции поддеревы за  $\mathcal{O}(\log n)$ .
- (b) 19. Euler-Tour Tree. Что это? Что умеет делать и за сколько?
- (b) 20. Euler-Tour Tree. Операции Link, Cut, IsConnected.
- (b) 21. Link-Cut. Описание структуры. Операция Expose.
- (b) 22. Link-Cut. Операции MakeRoot, Link, Cut.
- (c) 23. Link-Cut. Потенциал. Оценка времени Expose.
- (c) 24. Link-Cut. Оценка времени MakeRoot, *Link*, *Cut*.
- (c) 25. Link-Cut. Оценка времени со сплей.

## Паросочетания

- (a) 26. Matching. Определения и сложность задач в двудольном и произвольном графе: паросочетание, вершинное покрытие, независимое множество, совершенное паросочетание.
- (b) 27. Matching. Татт. Связь паросочетаний и определителей. Теорема Татта (без док-ва).
- (c) 28. Matching. Татт. Алгоритмы. Есть ли совершенное? *Найти max размер. Найти паросочетание.*

- (a) 29. Matching. Лемма о дополняющем пути.
- (a) 30. Matching. dfs для дополняющего пути, простейший алгоритм за  $\mathcal{O}(V(V+E))$ .
- (b) 31. Matching. Алгоритм Куна. Доказательство. Сравнение с обычным.
- (b) 32. Matching. Версия алгоритма Куна за  $\mathcal{O}(|M| \cdot E)$ . Оптимизации: обнуление пометок за  $\mathcal{O}(1)$ , жадная инициализация.
- (c) 33. Matching. Оптимизация алгоритма Куна «вообще не чистить пометки». Сравнение с предыдущими оптимизациями. *Связь с жадной инициализацией.*
- (a) 34. Matching. Алгоритм поиска VC и IS за  $\mathcal{O}(E)$ . Без док-ва.
- (b) 35. Matching. Теорема Кёнига и доказательство корректности алгоритма поиска VC и IS.
- (c) 36. Matching. Применение Куна: вероятностный алгоритм поиска паросочетания в  $\forall$  графе.
- (a) 37. Matching. *Разбиение вершин орграфа на циклы. Разбиение вершин ациклического орграфа на минимальное число путей.*
- (c) 38. Matching. *Теорема Дилворта, поиск максимальной антицепи.*
- (c) 39. Matching. Классификация рёбер. MAYBE = {рёбра, которые могут лежать в паросочетании}.
- (c) 40. Matching. Классификация рёбер. NO (не могут), MUST (должны).
- (a) 41. Matching. Stable matching (marriage problem). Постановка задачи. Решение. Пример про ЕГЭ.
- (b) 42. Matching. Stable matching. Корректность решения. Реализация за  $\mathcal{O}(V+E)$ .
- (b) 43. Matching. *Stable matching. Оптимальность для девочек, для мальчиков, единственность.*

## Потоки

- (a) 44. Flow. *Def:* поток, разрез, величина потока, величина разреза, остаточная сеть, циркуляция, прямые и обратные рёбра, декомпозиция потока.
- (a) 45. Flow. Алгоритм Форда-Фалкерсона. Время работы. Без доказательства.
- (b) 46. Flow. Теорема Форда-Фалкерсона. Доказательство.
- (a) 47. Flow. Поиск min разреза по max потоку за  $\mathcal{O}(E)$ .
- (b) 48. Flow. Эффективное хранение графа (интрузивный список, пары рёбер).
- (a) 49. Flow. Декомпозиция потока на пути за  $\mathcal{O}(E^2)$ . Циркуляция.
- (b) 50. Flow. *Декомпозиция потока за  $\mathcal{O}(VE)$ .*
- (a) 51. Flow. Решение задачи про  $k$  непересекающихся по рёбрам/вершинам путей.
- (a) 52. Flow. Поиск паросочетания через поток за  $\mathcal{O}(VE)$ .
- (c) 53. Flow. Паросочетание. Связь вершинного покрытия с разрезом. *Вершинное покрытие минимального веса.*
- (b) 54. Flow. Разности и суммы потоков. Леммы.
- (b) 55. Flow. Алгоритм Эдмондса-Карпа. Существование max потока при  $c_e \in \mathbb{R}$ . Время работы, часть доказательства.
- (b) 56. Flow. Алгоритм Эдмондса-Карпа. Доказательство леммы про рост расстояний.
- (b) 57. Flow. Алгоритм масштабирования потока. Время работы, доказательство.
- (c) 58. Flow. Алгоритм Диница. Время работы, доказательство.
- (c) 59. Flow. Алгоритм Диница с масштабированием. Время работы, доказательство.
- (c) 60. Flow. Алгоритм Диница с *link-cut-tree*. Время работы, доказательство.
- (c) 61. Flow. Алгоритм Хопкрофта-Карпа поиска паросочетаний.
- (c) 62. Flow. 1-я теорема Карзанова  $\mathcal{O}(\sqrt{C})$ .
- (c) 63. Flow. 2-я теорема Карзанова  $\mathcal{O}(U^{1/3}V^{2/3})$ .
- (b) 64. Flow.  $[L, R]$ -циркуляция за  $\mathcal{O}(flow)$ .
- (b) 65. Flow.  $[L, R]$ -flow,  $[L, R]$ -max-flow за  $\mathcal{O}(flow)$ .

- (a) 66. Flow. Глобальный разрез: определение, базовое решение за  $\mathcal{O}(V \cdot flow)$ .
- (b) 67. Flow. Глобальный разрез: алгоритм Штор-Вагнера без доказательства.
- (b) 68. Flow. Глобальный разрез: алгоритм Каргера-Штейна за  $\mathcal{O}(V^4)$ .
- (c) 69. Flow. Глобальный разрез: алгоритм Каргера-Штейна за  $\mathcal{O}(V^2 \log^2 V)$ .
- (a) 70. Flow. *Транспортная задача. Формулировка с  $\forall c_e$  и  $c_e = +\infty$ . Решение.*
- (a) 71. Flow. Задачи mincost  $k$ -flow, mincost max flow, mincost circulation. Сведения друг к другу.
- (a) 72. Flow. Алгоритм для поиска mincost  $k$ -flow в графе без отрицательных циклов за  $\mathcal{O}(k \cdot \text{FordBellman})$ . Без док-ва.
- (a) 73. Flow. Поиск паросочетания min/max веса через mincost потоки.
- (b) 74. Flow. Критерий оптимальности mincost  $k$ -flow. Доказательство.
- (b) 75. Flow. Алгоритм для поиска mincost  $k$ -flow. Корректность.
- (b) 76. Flow. Mincost  $k$ -flow в графе без отрицательных циклов за  $\mathcal{O}(\text{FordBellman} + k \cdot \text{Dijkstra})$ . Доказательство.
- (a) 77. Flow. Графы с отрицательными циклами. Алгоритм Клейна.
- (b) 78. Flow. ММСС. Алгоритм, *оценка времени работы* без доказательства.
- (c) 79. Flow. Capacity Scaling за  $\mathcal{O}(E \cdot \text{Dijkstra} \cdot \log U)$ .
- (c) 80. Flow. Задача mincost flow: *решения через итеративного Форд-Беллмана, через сведение к циркуляции.*

## Строки

- (a) 81. String. Префикс функция. КМП для поиска подстроки.
- (a) 82. String. Префикс функция. *Поиск всех периодов в строке.*
- (b) 83. String. LCP за  $\mathcal{O}(n^2)$ .
- (b) 84. String. Z-функция. Поиск подстроки в строке.
- (b) 85. String. Z-функция. *Поиск всех периодов строки.*
- (c) 86. String. Алгоритмы Бозера-Мура поиска подстроки в строке за сублинейное время.
- (a) 87. String. Хеши. Полиномиальный хеш, хеш подстроки за  $\langle \mathcal{O}(n), \mathcal{O}(1) \rangle$ .
- (a) 88. String. Хеши. Алгоритм Рабина-Карпа поиска подстроки в строке с  $\mathcal{O}(1)$  доппамяти.
- (a) 89. String. Хеши. Общие слова про хеширование. RP и ZPP версии Рабина-Карпа.
- (a) 90. String. Хеши. Каким хешом пользоваться и почему?
- (b) 91. String. Хеши. Связь с корнями многочленов, оценки  $\frac{|s|}{M}$  в худшем и  $\frac{1}{M}$  в среднем.
- (b) 92. String. Хеши. Вероятность ошибки для  $k$  сравнений подстрок хешами.
- (c) 93. String. Хеши. Оценки вероятности ошибки для Рабина-Карпа.
- (c) 94. String. Хеши. Оценки для двух вариантов «количества различных подстрок».
- (c) 95. String. Хеши. Антихеш тесты. Строка Туэ-Морса.
- (a) 96. String. Хеши. Наибольшая общая подстрока за  $\mathcal{O}(n \log n)$ .
- (b) 97. String. Хеши. *Поиск LCP и построение суффмассива за  $\mathcal{O}(n \log^2 n)$ .*
- (b) 98. String. *Поиск в тексте с одной ошибкой.*
- (c) 99. String. Палиндромы. *Подсчет числа за  $\mathcal{O}(n \log n)$ , самый длинный за  $\mathcal{O}(n)$ .*

## Бонус

- (+) 100. Splay. Статическая оптимальность (всё о ней). Гипотеза динамической оптимальности.
- (+) 101. Treap. Фишки. Что можно делать за  $\mathcal{O}(1)$ ? Почему это  $\mathcal{O}(1)$ ?
- (+) 102. RMQ.  $\langle \mathcal{O}(n), \mathcal{O}(1) \rangle$  одним махом.
- (+) 103. MST за линейное время в среднем. Собственно алгоритм.

- (+) 104. MST за линейное время в среднем. Лемма про количество «небесполезных рёбер».
- (+) 105. Matching. Задача о назначениях. Венгерский алгоритм, реализация за  $\mathcal{O}(V^3)$ .
- (+) 106. Flow. Push-relabel. Алгоритм за  $\mathcal{O}(V^3)$ . Корректность и времени работы.
- (+) 107. Flow. Push-relabel. Global Relabeling (без док-ва) и алгоритм Ahuja за  $\mathcal{O}(VE + V^2 \log U)$ .
- (+) 108. Flow. Cost Scaling: push-relabel для mincost потоков. Алгоритм Goldberg-Kennedy.