

Вопросы к экзамену по алгоритмам SPb HSE, 1-й курс ПАДИИ, март 2025

Общая информация

- Слайды от Данила: [\[link\]](#)
- Конспекты ПМИ: [\[part1\]](#) [\[part2\]](#) [\[part3\]](#)
- Практики и куски разборов: [\[wiki\]](#)
- Экзамен: письменный + устный, ≈ 1 час на подготовку билета, ≈ 20 минут на ответ записанного и дополнительных вопросов. Экзамен проходит *без* использования конспекта и других источников.
- *Курсивом* помечено то, что было разобрано на практике.
- (a) темы, обязательные к знанию (без них не получить удовл. оценку)
- (+) факультативные темы (оценка 10) – нужно заботать 7 **любых** темы из 10.

Графы, DSU и MST

- (a)
1. Алгоритм Краскала. Оценка времени работы.
 2. Краскал: корректность алгоритма (без ссылки на лемму).
 3. Алгоритм Прима. Оценка времени работы, вариации.
 4. Прим: корректность алгоритма (без ссылки на лемму).
 5. DSU на деревьях, две эвристики.
 6. DSU на деревьях: обоснование времени работы $\mathcal{O}(\log n)$.
 7. *Дерево кратчайших путей по массиву расстояний за $\mathcal{O}(E)$.*
 8. *Единственность MST для различных весов рёбер.*
 9. *Offline покраска на отрезке за $\mathcal{O}((n+m)\alpha)$.*

Структуры данных

- (a)
10. Дерево отрезков: базовая версия сверху для $n = 2^k$, sum/min.
 11. Нижняя оценка на время работы структуры, которая умеет change и getMin(l,r).
 12. Д.О. для хранения множеств и запросов «к-й элемент в множестве».
 13. Д.О. ленивое (динамическое) на массиве длины 10^{18} .
 14. *Дерево отрезков сортированных массивов.*
 15. *Дерево отрезков: k-ая статистика на отрезке за $\mathcal{O}(\log^3)$.*
 16. *Дерево отрезков: оптимизация DP для НВП $\maxLen[i]$ с n^2 до $n \log n$.*
 17. Дерево отрезков: присваивание на отрезке, прибавление на отрезке.
 18. RMQ: Sparse Table.
 19. LCA: Двоичные подъёмы (binary lifting).
 20. LCA: Минимум на пути в дереве за $\mathcal{O}(\log n)$.
 - (a) 21. BST. Определение, find, add, del, min/max всё за $\mathcal{O}(h)$.
 22. BST. Симметричный обход: сортировка деревом, дебаг вывод.
 23. BST. Что есть в C++? Чего нет?
 24. *BST. lowerBound за $\mathcal{O}(\log n)$.*
 25. *BST. next/prev за $\mathcal{O}(1)$; find за $\mathcal{O}(1)$; del за $\mathcal{O}(1)$.*
 26. *BST. Поиск точки в стакане: $l \leq x \leq r$ и $y \geq d$.*

- 27. AVL. Определение, оценка глубины.
- 28. AVL. Добавление, вращения, число вращений при добавлении.
- 29. AVL. Код малого вращения в 1 строку.
- 30. AVL. Ленивое удаление из AVL и других BST.
- 31. *xLR* обход дерева и хранение на диске.
- (a) 32. Операции $x \rightarrow$ позиция в BST; позиция в BST $\rightarrow x$.
- 33. Неявный ключ. Операции `Insert(i, x)` и `Delete(i)`.
- (a) 34. Treap: `Split/Merge`. `Insert/Delete`.
- 35. Treap и неявный ключ: операции `Split(i)/Merge`.
- 36. Treap: более быстрые `Insert/Delete`, через один `Split/Merge`.
- 37. Treap: оценка на среднюю глубину (с док-вом) и максимальную глубину (без док-ва).
- 38. Treap (rope): `reverse(l, r)`, `rotate(k)`.
- 39. Treap: операции на отрезке $\sum w_i$, $\min w_i$, модификации на отрезке ($+=$ на отрезке и $:=$ на отрезке).
- (a) 40. Персистентная версия BST/AVL/Treap.
- 41. Персистентность: частично-персистентный массив за $\mathcal{O}(1)$ на `change` и $\mathcal{O}(\log n)$ на `get`.
- 42. Персистентность: задача про копирование куска массива за $\mathcal{O}(\log n)$.
- 43. Персистентный массив в offline за $\mathcal{O}(1)$.
- 44. Персистентный массив в online за $\mathcal{O}(\log n)$.
- (a) 45. Scanline. 2D-запросы. Сумма в углу и сумма на прямоугольнике за $\mathcal{O}(\log n)$ в offline.
- 46. Scanline. Для каждой точки кол-во покрывающих её прямоугольников в offline.
- 47. Scanline. Персистентный. Количество чисел $\leq x$ на $[0..i]$.
- 48. Scanline. Персистентный. 2D-запросы в online за $\mathcal{O}(\log n)$.
- 49. Scanline. НВП и «возрастающая, максимальная по сумме весов»; за $\mathcal{O}(n \log n)$.
- 50. Scanline. Персистентный. k -ая статистика на отрезке за $\mathcal{O}(\log n)$ в online.
- 51. Эйлеров-обход. LCA за $\mathcal{O}(1)$.
- 52. Эйлеров-обход. Вычисление функции от поддерева.

Строки

- (a) 53. Определения: префикс, суффикс, период, подстрока. Срезы `s[:i]`, `s[i:]`, `s[i:j]`.
- (a) 54. Полиномиальные хеши: предподсчёт, хеш подстроки, выбор X и M .
- 55. Полиномиальные хеши: обоснование вероятности ошибки.
- 56. Алгоритм Рабин-Карпа с $\mathcal{O}(1)$ допамяти и оценка ошибки для него.
- 57. Количество различных подстрок за $\mathcal{O}(n^2)$ хешами. Оценка ошибки.
- 58. Сравнение строк на больше/меньше за $\mathcal{O}(\log n)$.
- 59. Наибольшая общая подстрока двух строк за $\mathcal{O}(n \log n)$.
- 60. Вычисление π -функции за $\mathcal{O}(n)$.
- 61. КМП: использование π -функции для поиска подстроки в строке.
- 62. Упражнение на π -функцию: найдите $\forall i$ число префиксов равных суффиксу `y s[:i]`.
- (a) 63. Бор: 3 варианта хранения.
- 64. Бор: сравнение вариантов хранения.
- (a) 65. Бор: поиск по словарю за $\mathcal{O}(|text| \cdot \max |s_i| + \sum |s_i|)$.
- 66. Бор: реализация `map<string, int>`.
- 67. Бор: сортировка строк; оценка времени работы.
- 68. Алгоритм Хаффмана сжатия текста.

- 69. Написание T9: подсказка наиболее вероятных слов по набранному префиксу.
- 70. Сжатый бор. Суффиксное дерево за $\mathcal{O}(n^2)$ и $\mathcal{O}(n)$ памяти.
- 71. Суффиксное дерево: поиск подстроки в строке за $\mathcal{O}(|s|)$.
- (a) 72. Суффиксный массив: определение, хранение в $\mathcal{O}(n)$ памяти.
- 73. Суффиксный массив за $\mathcal{O}(n \log^2 n)$ хешами.
- 74. Суффиксный массив за $\mathcal{O}(n \log n)$.
- 75. Суффиксный массив: поиск подстроки в строке за $\mathcal{O}(|s| \log |t|)$ и $\mathcal{O}(|s| + \log |s| \log |t|)$.
- 76. Суффиксный массив: подсчёт LCP за $\mathcal{O}(n)$.
- 77. Алгоритм Ахо-Корасик: построение суфф. ссылок.
- 78. Алгоритм Ахо-Корасик: построение полного автомата.
- 79. Алгоритм Ахо-Корасик: поиск всех словарных слов в тексте.
- 80. Z-функция.
- 81. Алгоритм Манакера поиска подстрок-палиндромов.

Корневая

- (a) 82. Кол-во $=x$ и $+=\delta$ на всём массиве за $\mathcal{O}(1)$.
- 83. Корневая по массиву: обобщение той же задачи до «на отрезке массива».
- 84. Корневая по массиву: кол-во $\geq x$ и $+=\delta$ на отрезке.
- 85. Корневая по массиву: сумма и изменение за $\langle \sqrt{n}, 1 \rangle$ и $\langle 1, \sqrt{n} \rangle$.
- (a) 86. Корневая по запросам: SortedArray и добавление новых элементов в online.
- 87. Корневая по запросам: «число точек в \square » и добавление/удаление новых точек online.
- 88. Корневая по длинам строк: «разбить текст на словарные слова» за $\mathcal{O}(|S| + |text| \sqrt{|S|})$.
- 89. Корневая по запросам: add/del ребра в граф + проверять связность (offline).
- 90. Корневая по запросам: остовное дерево минимальное по $\max w_e - \min w_e$.
- 91. Алгоритм Мо: количество различных на отрезке в offline.
- 92. Алгоритм Мо: количество инверсий на отрезке в offline.
- 93. Корневая split/rebuild: кол-во $=x$ на отрезке; $+=d$ на отрезке; вставка после i -позиции.

Random

- 94. Как и где уже пользовались рандомом? Q.HT.PH.FB.T.
- 95. Какие два типа рандомизированных алгоритмов есть? Как понизить вероятность ошибки?
- 96. Random Walk: 3-SAT формулировка и алгоритм за $\mathcal{O}^*(1.33^n)$.
- 97. Обоснование времени работы 1.73^n и 1.5^n для алгоритма выше.
- (a) 98. Monte-Carlo для площади пересечения/объединения кругов, шаров. Оценка погрешности.
- 99. Las-Vegas: на примере задачи «элемент массива, встречающийся $\leq \frac{n}{2}$ раз». 2 версии.
- 100. Поиск квадратичного невычета.
- (a) 101. Дерево 01-игры. Решение динамикой за $\mathcal{O}(2^n)$. Оптимизация рандомом.
- 102. Оценка оптимизации рандомом 1.686^n .
- 103. Поиск простого на отрезке.
- 104. Реализация RandomShuffle за $\mathcal{O}(n)$.
- 105. Решение мини-макс игры. В листьях произвольные вещественные числа. $\mathcal{O}^*(1.686^n)$.
- 106. 2-List-Coloring (жадное решение).
- 107. 3-List-Coloring (рандомизированное решение). Оценка времени работы.
- 108. Поиск двух ближайших точек на плоскости за $\mathcal{O}(n)$.

Дополнительные вопросы

- (+) 109. DSU на списках. Алгоритм Борувки (без оценки времени).
- (+) 110. RMQ за $\log^*(n)$ (Sparse Table на $\frac{n}{\log n}$ кусков) или быстрее.
- (+) 111. RB-дерево. И его AA-модификация.
- (+) 112. Splay-дерево и обоснование амортизированного времени работы $\mathcal{O}(\log B - \log A)$.
- (+) 113. Двоичные подъёмы с $\mathcal{O}(n)$ памяти.
- (+) 114. Центроидная декомпозиция и минимум на пути в дереве за $\mathcal{O}(1)$.
- (+) 115. Суб-линейный алгоритмы Боуэра-Мура для поиска подстроки в строке.
- (+) 116. Сортировка n строк над алфавитом n за $\mathcal{O}(n)$.
- (+) 117. Суффиксный массив за $\mathcal{O}(n)$.
- (+) 118. Рандомизированный алгоритм пересечения n полуплоскостей за $\mathcal{O}(n)$.