

Вопросы к экзамену по алгоритмам SPb HSE, 1-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 4 **любых** темы из 6.

База, асимптотики

- (a) 1. База. Асимптотика. Обозначения $o, O, \Theta, \Omega, \omega$.
- (a) 2. База. Основные свойства (7 штук). Асимптотика многочлена.
- (c) 3. База. Определение o, w, O через пределы.
- (a) 4. База. Связь умножения чисел и многочленов. Умножение за $O(n^2)$.
- (b) 5. База. Алгоритм Карацубы для умножения многочленов: идея, время работы.
- (a) 6. База. Рекуррентные соотношения: мастер-теорема. Формулировка.
- (b) 7. База. Доказательство мастер-теоремы.
- (b) 8. База. Теорема об экспоненциальном рекуррентном соотношении. Формулировка.
- (c) 9. База. Доказательство теоремы о рекуррентном соотношении.
- (b) 10. База. Доказательства по индукции. Пример $T(n) = \max_{x=1..n-1} (T(x) + T(n-x) + x(n-x))$.
- (c) 11. База. Доказательства по индукции. *Пример из дз: $T(n) = n + T(\frac{1}{2}n) + T(\frac{1}{3}n)$.*
- (a) 12. База. Числа Фибоначчи. Определение, способ и время вычисления.
Почему иногда говорим $\Theta(n)$, иногда говорим $\Theta(n^2)$?
- (a) 13. База. Примеры на асимптотику: 5 вложенных циклов, 2 указателя для $x^2 + y^2 = n$.
- (b) 14. База. Оценка суммы гармонического ряда через интеграл.
- (b) 15. База. Оценка суммы гармонического ряда через куски длины 2^k .
- (b) 16. База. Список делителей для всех чисел от 1 до n за $O(n \log n)$.
- (a) 17. База. Сравнение $n^2, 2^n, \log^2 n$. Только формулировки.
- (a) 18. База. Основы дебага: warnings, UB, glibcxxdebug.
- (a) 19. База. C++. Чем плохи стандартные ввод-вывод и new/delete?
- (b) 20. База. Неасимптотические оптимизации. Примеры 6 проблем.
- (c) 21. База. Кеш. Память. Особо медленные и особо быстрые операции.

Структуры данных

- (a) 22. DS. Частичные суммы.
- (a) 23. DS. Массив фиксированного размера. Что умеет? За сколько?
- (a) 24. DS. Список двусвязный, список односвязный, реализация на указателях.
- (b) 25. DS. Список односвязный, реализация на массиве.
- (a) 26. DS. Вектор (расширяющийся массив). Устройство.
- (b) 27. DS. Вектор. Доказательство времени работы.

- (a) 28. DS. Стек, очередь, дек. Описание.
- (b) 29. DS. Стек, очередь, дек. Описание. *Средства языка C++ и их сравнение.*
- (a) 30. DS. Очередь и дек через двусвязный список.
- (b) 31. DS. Очередь и дек на циклическом массиве.
- (c) 32. DS. Сравнение дека на списке и на циклическом массиве.
- (b) 33. DS. Стек с минимумом. Очередь с `min` через два стека.
- (c) 34. DS. *Очередь с минимумом, второй способ (дек минимумов).*
- (c) 35. DS. *Ближайший справа/слева меньший через стек.*
- (c) 36. DS. *`+=` на отрезке за $\mathcal{O}(1)$.*
- (a) 37. База. Потенциал. Время работы: реальное, амортизировано, суммарное, среднее.
- (a) 38. База. Связь амортизированного времени и реального. Случай $\varphi \geq 0$.
- (b) 39. База. Связь амортизированного времени и реального. Произвольный случай с $\Delta\varphi = \varphi_{end} - \varphi_0$.
- (b) 40. База. Примеры на потенциалы: `push/pop(k)`, хороший и плохой потенциал для $x^2 + y^2 = n$
- (c) 41. База. Схема с монетками. Связь с обычными потенциалами. Пример для вектора.
- (b) 42. DS. Разбор арифметического выражения со стеком за линейное время.
- (a) 43. DS. Бинпоиск: трёхветочный, свой `lowerbound`, средства языка C++.
- (b) 44. DS. Бинпоиск: по предикату, `lowerbound` и `upperbound` через предикат.
- (a) 45. DS. Бинпоиск вещественный: для монотонной функции, для корня многочлена 3-й степени.
- (a) 46. DS. Бинпоиск: определение границ L, R бинпоиска для корня многочлена 3-й степени.
- (c) 47. DS. *Бинпоиск: корни многочлена \forall степени.*
- (c) 48. DS. *Бинпоиск: минимум массива $a_1 > a_2 > \dots > a_k < \dots < a_n$.*
- (a) 49. DS. Два указателя. Хранение множеств и мультимножеств в виде сортированных массивов. Пересечение, объединение, разность за $\mathcal{O}(n)$. Средства языка C++.
- (a) 50. DS. *Два указателя. 3-SUM за $\mathcal{O}(n^2)$: найти $i, j, k: a_i + a_j + a_k = S$.*
- (a) 51. DS. *Два указателя. Максимальный отрезок, без повторяющихся чисел.*
- (b) 52. DS. *Два указателя. Минимальный отрезок, содержащий k различных чисел.*
- (b) 53. DS. *Два указателя. Отрезок максимальной суммы длины от A до B .*
- (a) 54. DS. Хеш-таблица. Версия на списках. Средства языка C++.
- (b) 55. DS. Хеш-таблица. Версия с открытой адресацией.
- (a) 56. DS. *События. Для каждой точки найти «количество покрывающих её отрезков».*
- (b) 57. DS. *События. Покраска отрезков массива за $\mathcal{O}(n + q \log q)$.*
- (c) 58. DS. *Расширяющийся и сужающийся дек/динамический массив. Доказательство времени работы.*
- (b) 59. DS. Вектор: избавление от амортизации. Любой способ.
- (c) 60. DS. Вектор: избавление от амортизации. Два способа.
- (c) 61. DS. Хеш-таблица с открытой адресацией: избавление от амортизации.
- (c) 62. DS. Очередь с минимумом: избавление от амортизации.
- (a) 63. Неар. Бинарная куча: хранение в массиве, `add`, `extractMin`.
- (a) 64. Неар. Обратные ссылки. Что это, зачем нужны?
- (b) 65. Неар. Задача `a[i]=x`, `getMin`.
- (b) 66. Неар. `decreaseKey`, `delete`.
- (b) 67. Неар. Heapsort (inplace). Средства языка C++.
- (c) 68. Неар. Построение за линию (алгоритм, оценка).
- (c) 69. Неар. *Оценка снизу: нет кучи, умеющей и `add`, и `extractMin` за $o(\log n)$.*
- (a) 70. DS. Аллокация памяти. Стек.

- (b) 71. DS. Аллокация памяти. *Переопределение new, delete на стековый.*
- (b) 72. DS. Аллокация памяти. Список.
- (c) 73. DS. Аллокация памяти. Куча. Описание структуры без подробностей реализации.
- (b) 74. DS. Ничего \rightarrow Del. Пример. Куча.
- (a) 75. DS. Find \rightarrow Del. Пример. Хеш-таблица.
- (a) 76. DS. Add \rightarrow Merge. Пример. Куча.
- (b) 77. DS. Add \rightarrow Merge. Оценка времени работы
- (b) 78. DS. Пополняемый массив. Build \rightarrow Add. Корневая.
- (c) 79. DS. Пополняемый массив. Build \rightarrow Add. $\mathcal{O}(\log^2 n)$.
- (c) 80. DS. Build \rightarrow Add. Оценка для произвольной структуры.
- (b) 81. DS. Build, Add \rightarrow Del.
- (c) 82. DS. Алгоритм Мо. Два указателя на примере задачи «количество различных чисел на отрезке». Собственно алгоритм.
- (c) 83. DS. Алгоритм Мо. Подробная оценка времени работы, случай $n \neq m$.

Сортировки

- (a) 84. Sort. Квадратичные: алгоритмы Selection, Insertion, Bubble;
- (a) 85. Sort. Стабильность. Число инверсий (определение).
- (b) 86. Sort. Insertion(+BS), время работы, число сравнений.
- (b) 87. Sort. Подробный анализ/сравнение сортировок за квадрат: число сравнений, присваиваний, стабильность, допамять.
- (a) 88. DS. Задача про пересечение A и B . Три решения.
- (a) 89. Sort. Оценка снизу на число сравнений.
- (c) 90. Sort. *Оценка снизу: обобщения из практики и дз (на $\frac{1}{100}$ всех, на $\frac{1}{2^n}$).*
- (a) 91. Sort. MergeSort: рекурсивная версия.
- (b) 92. Sort. MergeSort: нерекурсивная версии.
- (b) 93. Sort. *MergeSort: подсчёт числа инверсий.*
- (a) 94. Sort. QuickSort. Простейший partition на $< x, = x, > x$.
- (a) 95. Sort. QuickSort. Inplace partition. Способы выбора элемента для partition.
- (b) 96. Sort. QuickSort. Доказательство через дерево рекурсии и вероятность «сравнить 2 элемента».
- (c) 97. Sort. QuickSort. Доказательство по индукции с интегралами.
- (c) 98. Sort. QuickSort. $\log n$ допамяти в худшем.
- (a) 99. Sort. IntroSort.
- (a) 100. Sort. Порядковая статистика за $\mathcal{O}(n)$, рандомизированный алгоритм. Средства C++.
- (b) 101. Sort. Порядковая статистика за $\mathcal{O}(n)$, детерминированный алгоритм. Алгоритм.
- (c) 102. Sort. Порядковая статистика за $\mathcal{O}(n)$, детерминированный алгоритм. Оценка времени.
- (a) 103. Sort. CountSort для чисел.
- (b) 104. Sort. CountSort для пар чисел.
- (c) 105. Sort. Radix Sort за $\mathcal{O}(n \log_n m)$.
- (a) 106. Sort. Bucket Sort. Описание алгоритма.
- (b) 107. Sort. Bucket Sort. Две версии алгоритма. Время работы (формулировка).
- (c) 108. Sort. Bucket Sort. Две версии алгоритма. Время работы (доказательства).

Кучи

- (b)109. Heap. Van Emde Boas trees. `add` за $\mathcal{O}(\log \log C)$.
- (b)110. Heap. Van Emde Boas trees. `extractMin` за $\mathcal{O}(\log \log C)$.
- (c)111. Heap. Van Emde Boas trees. *lowerBound*.
- (c)112. Heap. Van Emde Boas trees, где тут хеш-таблицы? Реализация на одной хеш-таблице.
- (a)113. Heap. MinMax heap. Хранение.
- (b)114. Heap. MinMax heap. Алгоритм `siftUp`, `siftDown`.
- (c)115. Heap. MinMax heap. Подробная оценка времени работы, сравнение с бинарной.
- (a)116. Heap. Leftist heap. `Merge`, `Add`, `extractMin`. Только алгоритм.
- (b)117. Heap. Leftist heap. Обоснование времени работы.
- (b)118. Heap. Skew Heap: операции `add`, `extractMin`.
- (b)119. Heap. Лёгкие и тяжёлые рёбра, доказательство времени работы Skew Heap.
- (c)120. Heap. Сравнение куч: Leftist heap. Skew heap. Бинарная.
- (c)121. Heap. *Skew Heap: delete, decreaseKey. Равносильность.*
- (b)122. Heap. *d-куча.*
- (a)123. Heap. Списко-куча, умеющая $\{\text{add, min, merge, decreaseKey}\}$ за $\mathcal{O}(1)$ и `extractMin` за $\mathcal{O}(n)$.
- (b)124. Heap. Турнирные деревья. Операции `change(i, x)` и `extractMin` за $\mathcal{O}(\log n)$.
- (b)125. Heap. Список турнирный деревьев: `add`, `min`, `merge` за $\mathcal{O}(1)$, `decreaseKey` и `extractMin` за $\mathcal{O}(\log n)$.
- (c)126. Heap. QuakeHeap. `decreaseKey` за $\mathcal{O}(1)$, землетрясения.
- (c)127. Heap. QuakeHeap. Доказательство времени `extractMin` $\mathcal{O}(\log n)$.
- (c)128. Heap. Нижняя оценка на время построения бинарной кучи.
- (a)129. Inplace. *Unique, Reverse* за $\mathcal{O}(n)$. Средства языка C++.
- (b)130. Inplace. *Rotate* за $\mathcal{O}(n)$. Средства языка C++.
- (b)131. D&C. *Поиск двух ближайших точек.*
- (c)132. D&C. *Stable Inplace Merge* за $\mathcal{O}(n \log n)$.

- (+)133. База. Сравнение полинома и экспоненты по индукции. Доказательство.
- (+)134. DS. Аллокация памяти. Куча. Подробная реализация.
- (+)135. Sort. Adaptive Heap Sort.
- (+)136. Sort. Kirkpatrick Sort за $\mathcal{O}(n \log \log C)$.
- (+)137. Inplace. `Merge` за $\mathcal{O}(n)$.
- (+)138. Heap. Pairing Heap.