

# Вопросы к экзамену по алгоритмам

## СПб АУ, первый курс, весна, 2017/18 учебный год

Кроме конспектов полезно смотреть **разборы** задач с практик и из дз.

### Необходимые знания из прошлого

- Рекуррентности: мастер-теорема.
- Амортизированное время работы: определение, теорема, оценка для вектора.
- Бинпоиск. Сортировки heapsort/mergesort/quicksort.
- Хеш-таблицы на списках, с открытой адресацией.
- НОП за  $\mathcal{O}(n^2)$ . НВП за  $\mathcal{O}(n \log n)$ .
- Рюкзак со стоимостями (knapsack) за  $\mathcal{O}(nW)$ .
- Гамильтонов путь и коммивояжер за  $\mathcal{O}(2^n n^2)$ . Раскраска вершин графа за  $\mathcal{O}(3^n)$ .
- dfs. Поиск компонент связности, поиск цикла в орграфе.
- Определения NP, NPc, ВН, SAT, RP, BPP, понижение ошибки для RP и BPP.
- Дейкстра за  $\mathcal{O}(E \log V)$ . Форд-Беллман за  $\mathcal{O}(VE)$ .
- MST. Краскал, Прим.

### Деревья поиска

- (a) 1. BST. Определение, add/del/prev/next/lower\_bound, использование списка и хеш-таблицы. Обработка равных ключей.
- (a) 2. BST. Персистентность. Персистентные операции “добавление” и “удаление”.
- (a) 3. BST. AVL-дерево. Инвариант, single/double rotation, add/del. Число вращений при add/del.
- (b) 4. BST. AVL-дерево. Split/Merge. Persistent AVL. Неявный ключ.
- (b) 5. BST. B-дерево: find, add/del, split/merge. B\*-дерево и B<sup>+</sup>-дерево. Неявный ключ.
- (a) 6. BST. 2-3-дерево, 2-3-4-дерево, RB-дерево, AA-дерево. Неявный ключ.
- (b) 7. BST. RBST (без  $y$ ), Treap (с  $y$ ). Split/merge. Два способа для add/del. Персистентная версия.
- (b) 8. BST. Splay-дерево: add/del, split/merge, потенциал, формулировки теорем.
- (b) 9. BST. Splay-дерево: доказательство общей теоремы о времени работы и теоремы о времени работы с учётом частот.
- (a) 10. BST. Запрос на отрезке, модификация на отрезке, reverse на отрезке.

## Деревья отрезков и другие структуры

- (a) 11. Центроидная декомпозиция: построение, минимум на пути.
- (b) 12. Центроидная декомпозиция: покраска вершин на расстоянии  $\leq d$ , число путей длины  $\leq d$ .
- (b) 13. Rope: интерфейс, возможные реализации. Skip-List: add/del, split/merge.
- (a) 14. Sqrt декомпозиция. По массиву (с примером), по запросам (с примером).
- (b) 15. Sqrt декомпозиция. Split/rebuild (с примером), split/merge (с примером), выбор  $k$ .
- (a) 16. Sqrt декомпозиция по запросам. Dynamic sorted array. Dynamic Connectivity Offline.
- (a) 17. Персистентность. Подход к любой структуре (примеры: СММ, хеш-таблица); offline-решение.
- (b) 18. Персистентная стек и очередь с операциями за  $\mathcal{O}(1)$ .
- (a) 19. Персистентная дек с операциями за  $\mathcal{O}(\log n)$ .
- + 20. Персистентность через fat-nodes. Применение в вычислительной геометрии.
- (a) 21. ДО. С операциями снизу (все оценки, код).
- (a) 22. ДО. С операциями сверху (все оценки, сравнение с реализацией снизу).
- (a) 23. ДО. Динамическое; сжатие координат; пример использования обеих идей в scanline.
- (b) 24. ДО. Многомерные деревья: сортированный массив, treap, ДО, 2D ДО.
- (a) 25. ДО. Scanline: число точек в прямоугольнике, число прямоугольников, покрывающих точку. Персистентная версия scanline для обеих задач.
- (a) 26. ДО. Площадь объединения прямоугольников.
- (b) 27. ДО.  $k$ -я статистика на отрезке за  $\mathcal{O}(\log^3 n)$ ,  $\mathcal{O}(\log^2 n)$ ,  $\mathcal{O}(\log n)$ .
- + 28. Fractional-cascading, КД-дерево.

## LCA, RMQ

- (a) 29. RMQ. Формулировка задачи. Lower bound. Решение через ДО, корневую, Sparse Table.
- (b) 30. RMQ. Замыкание Sparse Table до  $\langle n \log^* n, \log^* n \rangle$ . Disjoint Sparse Table.
- (a) 31. LCA. Формулировка, решение двоичными подъёмами, функция на пути дерева.
- (a) 32. RMQ $\pm 1$  за  $\langle \mathcal{O}(n), \mathcal{O}(1) \rangle$ .
- (a) 33. LCA за  $\langle \mathcal{O}(n), \mathcal{O}(1) \rangle$ . Три типа эйлерового обхода, пример применения каждого.
- (b) 34. RMQ за  $\langle \mathcal{O}(n), \mathcal{O}(1) \rangle$ . Построение декартова дерева за  $\mathcal{O}(n)$ .
- (a) 35. LCA в offline. Алгоритм Тарьяна.
- (b) 36. LA. Решение в offline. Алгоритм Вишкина.
- + 37. LA. Реализация Вишкина за  $\langle \mathcal{O}(n \log n), \mathcal{O}(1) \rangle$ . Решение за  $\langle \mathcal{O}(n), \mathcal{O}(1) \rangle$ .
- (a) 38. ЕТТ. Euler Tour Tree. Функция от поддерева, модификация в поддереве.
- (b) 39. HLD. Heavy Light Decomposition. Функция на пути дерева, модификация на пути дерева.
- (b) 40. LCT. Link Cut Tree. Описание структуры, введение потенциала, оценка Expose.
- (b) 41. LCT. Link Cut Tree. Оценка MakeRoot, Link, Cut.
- + 42. LCT. Link Cut Tree. Оценка Expose и MakeRoot со Splay-деревом.
- (b) 43. MST за  $\mathcal{O}(n + m)$ . Оценка  $\min(m \log n, n^2)$  в худшем.
- + 44. RMQ в offline через алгоритм Тарьяна.

## Правила сдачи

Вы записываетесь в [google-doc](#), получаете время сдачи, когда нужно прийти и сдать. В нужный момент вы приходите и говорите “я готов сдавать!”. Если есть свободный принимающий, в ответ вы получаете **три случайных числа** – номера вопросов в каждой из групп, и идёте **20 минут готовиться**. Если вы опоздали более чем на 5 минут, попадаете в живую очередь таких же опоздавших, у вас самый низкий приоритет, но как только кто-то осводится, вас тоже послушают.

### При ответе билетов:

1. В **любых** спорных ситуациях подходим к лектору.
2. Лучше рассказать хоть что-то, чем ничего не сказать.
3. Вы готовились к экзамену! Вы должны помнить, что вы должны знать, что к какому билету относится.
4. Если вас спрашивают что-то не из билета, не из программы, не стесняйтесь говорить об этом.
5. Принимающий может в любой момент задать не сложный допвопрос из серии “что такое дек?” или “чем MergeSort лучше QuickSort?”.
6. Если вы плохо знаете свой билет, можно вкратце проговорить, что вы всё-таки по теме помните, и заменить билет. Замена происходит так: принимающий мгновенно генерит псевдослучайное число отличное от предыдущего и говорит вам. Замена билета – минус к оценке, зависящий от сложности билета.
7. Заменять билет можно только один раз.
8. Вы получаете 3 билета. По каждому у принимающего магическим образом складывается в голове оценка от 0 до 1 – уровень ваших знаний. Ваша оценка – сумма этих чисел, число от 0 до 3. Какие из этих чисел сообщать вам, а какие нет, решает принимающий. Вывод о ваших знаниях принимающий может описать вам не числом, а лишь на словах “тут вы почти ничего не знаете”, “этот билет вы ответили идеально”.

### Алгоритм выставления оценки:

- a) Меньше 1.33 – двойка. Ваши знания не удовлетворительны.
- b) Больше 2.66 – пятёрка. Вы круты.
- c) 2.2 и больше – точно четвёрка.
- d) 1.8 и меньше – точно тройка.
- e) Если от 1.8 до 2.2, то идёт серия простых и не очень допвопросов, помогающих уточнить оценку.