

# Вопросы к коллоквиуму по алгоритмам

## СПБ АУ, первый курс, весна, 2014/15 учебный год

### Деревья поиска

1. (a) Дерево поиска: find, add, del за  $\mathcal{O}(h)$ . Хранение дерева. Обходы дерева: симметричный, прямой.
2. (a) Дерево поиска: next, prev, find за  $\mathcal{O}(1)$ . Использование двусвязного списка, хеш-таблицы.
3. (a) Случайное дерево поиска. Два эквивалентных определения. Оценка максимальной (формулировка) и средней (с доказательством) глубины.
4. (a) AVL дерево: инвариант, оценка размера и глубины, малые и большие вращения, find, add, del.
5. (b) AVL дерево: split за  $\mathcal{O}(\log^2 n)$ , merge за  $\mathcal{O}(\log n)$ .
6. (a) Treap: единственность, глубина, split, merge, add, del за  $\mathcal{O}(\log n)$ ; build за  $\mathcal{O}(n \log n)$ .
7. (b) Rope. Неявный ключ: идея, применение к AVL, treap. Sum, min/max, add, reverse на отрезке.
8. (b) Skip-List (add, del, k-th, split, merge, всё за  $\mathcal{O}(\log n)$ ).
9. (b) Persistent Tree, Persistent Array, ссылочный garbage collector, разбор задачи про “копирование памяти”
10. (c) Persistent Stack, Queue за  $\mathcal{O}(1)$ .
11. (b) RBST: split, merge, add, del за  $\mathcal{O}(\log n)$ ; build за  $\mathcal{O}(n)$ .
12. (b) 2-3-tree, 2-3-4-tree, B-tree, RB-Tree, AA-Tree.
13. (b) Статически оптимальное дерево поиска. Определение. Построение за  $\mathcal{O}(n^2)$ .

### Дерево отрезков, RMQ

14. (a) Дерево отрезков с операциями сверху, модификация в точке, на отрезке, динамическое дерево отрезков для диапазонов произвольной длины. Обработка запросов add(x), del(x), count(l, r) деревом отрезков.
15. (b)  $k$ -я порядковая статистика, static data, online,  $\langle \mathcal{O}(n \log n), \mathcal{O}(\log n) \rangle$ .
16. (c)  $k$ -я порядковая статистика, dynamic data, offline,  $\langle \mathcal{O}(n \log n), \mathcal{O}(\log^2 n) \rangle$  (и online за  $\mathcal{O}(\log n \cdot \log M)$ ).
17. (b) 2D-orthogonal-query. Решение в online для ассоциативной функции за  $\langle \mathcal{O}(n \log n), \mathcal{O}(\log^2 n) \rangle$  (дерево отрезков сортированных массивов, дерево merge-sort-a). Улучшение решения до  $\langle \mathcal{O}(n \log n), \mathcal{O}(\log n) \rangle$  и объяснение, для каких функций это работает.
18. (b) Сведение задач двумерного поиска на массиве и двумерного поиска на плоскости друг к другу. Обработка  $d$ -мерного запроса деревом отрезков за  $\mathcal{O}(\log^d n)$ .
19. (a) Дерево отрезков с операциями снизу. Дерево Фенвика. Многомерное дерево Фенвика.
20. (a) Сканирующая прямая. 2D-orthogonal-query в offline для обратимых функций за  $\langle \mathcal{O}(n), \mathcal{O}(n \log n) \rangle$ . Максимальная по весу возрастающая по обеим координатам последовательность точек за  $\mathcal{O}(n \log n)$ .
21. (a) Сканирующая прямая. Площадь объединения прямоугольников.
22. (a) Sparse Table, Disjoint Sparse Table, частичные суммы, полный подсчёт. Классификация функций: какие функции какими из перечисленных структур считаются.
23. (c) Модификации Sparse Table и Disjoint Sparse Table:  $\langle \mathcal{O}(n), \mathcal{O}(\log n) \rangle$ ,  $\langle \mathcal{O}(n \log \log n), \mathcal{O}(1) \rangle$ ,  $\langle \mathcal{O}(n), \mathcal{O}(\log \log n) \rangle$ ,  $\langle \mathcal{O}(n \log^* n), \mathcal{O}(\log^* n) \rangle$ .
24. (c) Алгоритм Фараха-Колтона-Бендера: последовательность сведений и решение задачи  $RMQ_{\neq 1}$  за  $\langle \mathcal{O}(n), \mathcal{O}(1) \rangle$
25. (b)  $RMQ \rightarrow LCA$ . Построение Cartesian Tree за  $\mathcal{O}(n)$ .
26. (b)  $LCA \rightarrow RMQ$ . Эйлеров обход дерева (как минимум 3 версии). Применение: функция на поддереве.

## ЛСА, Функции на путях дерева

27. (b) Алгоритм Тарьяна: LCA-Offline. Модификация алгоритма: RMQ-Offline.
28. (a) Двоичные подъёмы, решение LCA за  $\langle \mathcal{O}(n \log n), \mathcal{O}(\log n) \rangle$ . Разреженная версия за  $\langle \mathcal{O}(n), \mathcal{O}(\log n) \rangle$ . Функции на путях не меняющегося дерева в online за  $\mathcal{O}(\log n)$ .
29. (a) Функции на путях дерева в offline. Решение dfs-ом и деревом отрезков. Функции на путях дерева: биекция между вершинами и рёбрами.
30. (a) Сумма на путях дерева в online. Случай неменяющегося дерева:  $\mathcal{O}(n), \mathcal{O}(1)$ . Случай меняющегося дерева:  $\mathcal{O}(n), \mathcal{O}(\log n)$ .
32. (c) Рандомизированный алгоритм построения MST за  $\mathcal{O}(n + m)$ . При раскраске билета считается, что мы умеем за  $\mathcal{O}(1)$  в offline находить минимум на пути дерева.
33. (b) Euler-Tour-Trees (ET): link, cut, makeRoot, isConnected за  $\mathcal{O}(\log n)$ .
34. (b) Heavy-Light-Decomposition (HLD): build за  $\mathcal{O}(n)$ , get за  $\mathcal{O}(\log^2 n)$ .
35. (b) Задача LA. Решение в offline за  $\mathcal{O}(n + m)$ . Решения в online двоичными подъёмами, разреженными двоичными подъёмами, HLD.
36. (c) Задача LA. Longest-Path-Decomposition, Ladder-Decomposition, решение за  $\langle \mathcal{O}(n \log n), \mathcal{O}(1) \rangle$ . Идея: хранить можно только листья. Микро-макро эвристика, решение за  $\langle \mathcal{O}(n), \mathcal{O}(1) \rangle$ .

## Дополнительный вопрос

31. (c) Минимум на путях дерева в offline за  $\mathcal{O}^*(1)$ .
- 

## 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 отвечать **без подготовки**. 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 спорных ситуациях подходим к лектору.
2. Лучше рассказать хоть что-то, чем ничего не сказать.
3. Вы готовились к коллоквиуму! Вы должны хотя бы помнить, что вы должны знать, и что к какому билету относится. Если вас спрашивают что-то не из билета, не из программы, говорите об этом.
4. Принимающий может в любой момент задать не сложный допвопрос из серии “что такое Disjoint Sparse Table?” или “кто такие orthogonal queries?”.
5. 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.
6. Заменять билет можно только один раз.
7. По каждой из тем 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 решает принимающий. Сделанный вывод принимающий может описать не числом, а лишь на словах “тут вы почти ничего не знаете”, “а этот билет вы ответили идеально”.

### Алгоритм выставления оценки:

1. Меньше 1 – двойка. Ваши знания не удовлетворительны.
2. Больше 2.5 – пятёрка. Вы круты.
3. 2 и больше – точно четвёрка.
4. 1.5 и меньше – точно тройка.
5. Если от 1.5 до 2, то идёт серия простых и не очень допвопросов.