

# Вопросы на 3 к экзамену по алгоритмам SPb HSE, 1 курс, июнь 2022

Собрано 28 августа 2022 г. в 12:15

## Жадности и приближённые алгоритмы

- (a) 1. Алгоритм Хаффмана, доказательство, реализация кучей за  $\mathcal{O}(k \log k)$ .
- (a) 2. Жадность. Задачи на сортировку. Способы доказательства с примерами задач: через swap соседних и через выбор первого/последнего элемента.
- (a) 3. Жадность. Задачи на сортировку. Примеры: выполнить побольше задач с общим дедлайном, файлы на ленте, *непрерывный рюкзак, башня из спортсменов*.
- (a) 4. TSP. 2-ОПТ решение.
- (a) 5. Рюкзаки. Формулировки: partition, knapsack, bin packing.
- (a) 6. Рюкзаки. В предположении NP-трудности partition показать трудность knapsack и bin packing.
- (a) 7. Partition. Жадное решение (без доказательства оценки).
- (a) 8. Partition. Алгоритм Кармаркар-Карпа (без доказательства оценок).
- (a) 9. Bin Packing. *Доказательство того, что First Fit, Next Fit – это 2-приближения.*
- (a) 10. Set Cover.  $\ln(n)$ -приближение. Решение без доказательства.

## Центроиды

- (a) 11. Центроидная декомпозиция: построение за  $\mathcal{O}(n \log n)$  времени,  $\mathcal{O}(n)$  памяти.

## Бинарные деревья поиска

- (a) 12. BST. Add, Del, Find, LowerBound за  $\mathcal{O}(h)$ , симметричный обход. *Нижняя оценка на Add.*
- (a) 13. BST. Next, Prev, Find за  $\mathcal{O}(1)$ . Итератор в `std::set`.
- (a) 14. AVL. Инвариант, оценка глубины. Add, перебалансировка.
- (a) 15. Общие идеи: BST и неявный ключ, BST и персистентность.
- (a) 16. Treap. Два определения случайного дерева (RBST). Эквивалентность определений. Treap является RBST.
- (a) 17. Treap. Матожидание глубины RBST (без доказательства) и глубины любого элемента RBST (с доказательством).
- (a) 18. Treap. Операции Split, Merge. *Оценка времени работы.* Неявный ключ.

## Структуры данных

- (a) 19. Persistent. Массив – любой способ.
- (a) 20. Persistent. Offline, дерево версий. *Детская персистентность для массива (2 версии), частичная персистентность за  $\mathcal{O}(\log n)$ .*
- (a) 21. Persistent. Стек.
- (a) 22. Дерево отрезков. Реализация сверху. Оценка на память, на время get.
- (a) 23. Дерево отрезков. Массовые модификации. += и getMin. = и getSum.
- (a) 24. Scanline. Offline: число точек в прямоугольниках за  $\mathcal{O}(\log n)$ ; *число прямоугольников, накрывающих точку за  $\mathcal{O}(\log n)$ .*
- (a) 25. 2D. ДО из сортированных массивов. Связь с двумерным запросом (количеством точек в прямоугольнике за  $\mathcal{O}(\log^2 n)$ ).

## RMQ, LCA, LA

- (a) 26. RMQ. Разреженная таблица (Sparse Table).
- (a) 27. RMQ. Эйлера обход дерева. Запросы на поддеревьях.
- (a) 28. RMQ. Сведение LCA  $\rightarrow$  RMQ $\pm$ 1. Решение задачи LCA за  $\langle \mathcal{O}(n \log n), \mathcal{O}(1) \rangle$ .
- (a) 29. LCA. Двоичные подьёмы. Решение LCA, два способа (с одной двигающейся вершиной и с двумя).
- (a) 30. Сумма на пути в дереве за  $\langle \mathcal{O}(n), \mathcal{O}(1) \rangle$ .
- (a) 31. LA. Решение LA в online за  $\langle \mathcal{O}(n \log n), \mathcal{O}(\log n) \rangle$  и в *offline* за  $\langle \mathcal{O}(n), \mathcal{O}(1) \rangle$ .