

Вопросы к экзамену по алгоритмам

SPb HSE, ПАДИИ 1-й курс, 17 июня 2024

Общая информация

- Конспекты ПМИ: [\[part1\]](#) [\[part2\]](#) [\[part3\]](#)
- Практики и куски разборов: [\[wiki\]](#)
- Слайды от Данила: [\[link\]](#)
- 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.

Хеш-таблицы и хеширование

1. Хеш-функция. Что это? Основные свойства.
Где, для чего используются хеш-функции? (хотя бы два примера).
2. Полиномиальные хеши строк: сформулируйте задачу, которую мы решаем, опишите предподсчёт, который мы делаем, оцените время работы.
3. Полиномиальные хеши строк: оценка вероятности коллизий.
4. Полиномиальные хеши строк: выбор пары $\langle x, mod \rangle$, обоснование.
5. Полиномиальные хеши строк: сравнение строк на больше-меньше за $\mathcal{O}(\log n)$.
6. Алгоритм Рабина-Карпа поиска подстроки в строке с $\mathcal{O}(1)$ памяти.
7. Хеш-таблица, как интерфейс. Что и за сколько должна уметь хеш-таблица?
8. Хеш-таблица на списках (цепочечная, закрытая). `add/del/find`.
9. Хеш-таблица с открытой адресацией. `add/find`.
10. Хеш-таблица с открытой адресацией. `del`.
11. Хеш-таблица с открытой адресацией: двойное хеширование.
12. Сравнение открытой и списочной реализаций хеш-таблицы (см. конспект ПМИ).
13. Хеш-таблица. Перехеширование. Что делать, если количество хранимых элементов заранее не известно? Доказательство амортизированного времени работы `add/find` $\mathcal{O}(1)$.

Хеш-таблицы и хеширование, продолжение

14. Хеширование Кукушки. `add/del/find`. Описание алгоритма, сравнение с уже известными хеш-таблицами.
15. Универсальное семейство хеш-функций: определение, пример универсального семейства, использование в хеш-таблицах, выбор случайной хеш-функции.
16. Ассоциативный массив. Что это? Схожесть с массивом, отличие от массива. Простой вопрос.
17. Многомерный ассоциативный массив. Две разных реализации для k -мерного ассоциативного массива: простая, эффективная.
18. Multiset. Multimap. Связь multimap с set. Отличие multiset от set.
19. Криптохеш. Основные свойства (4 свойства: эффект лавина и другие). Контрольная сумма. Любые два примера использования криптохешей.
20. Что такое симметричное шифрование?
Общая схема блочного алгоритма шифрования на примере SHA-256.
21. Фильтр-Блума. Описание алгоритма.
Зависимость вероятности ошибки от n, m, k (растёт? убывает?).
22. Фильтр-Блума. Примеры использования: любые два примера (есть два про IP-адреса; кеширование сайтов; есть ещё).

Строки: база

23. Префикс-функция. Определение, построение за $\mathcal{O}(n)$.
24. КМП для поиска подстроки p в строке s . Версия $\mathcal{O}(|p|)$ допамяти.
25. Детерминированный конечный автомат. Определение. Принимает ли автомат строку?
26. Поиск строки s в тексте t по полному автомату s за $\mathcal{O}(|t|)$.
27. Построение полного автомата s за $\mathcal{O}(|s| \cdot |A|)$, где A – алфавит.
28. Z-функция. Определение, построение за $\mathcal{O}(n)$.
29. Поиск периода строки.
30. Наибольшая общая подстрока двух строк без суффиксных структур.
31. Бор. Способы хранения (массив, `map`, `unordered_map`, сравнение).
32. Бор. Примеры использования: сортировка строк, `unordered_map<string, int>`.

Строки: продвинутое

33. Сжатый бор. Определение и построение за $\mathcal{O}(n^2)$ суффиксного дерева. Что сделать, чтобы все суффиксы заканчивались в листьях?
34. Применения суффиксного дерева: поиск подстроки в тексте, общая подстрока.
35. Алгоритм Ахо-Корасик. Построение полного автомата за $\mathcal{O}(|A| \cdot \sum_i |s_i|)$, для алфавита A .
36. Алгоритм Ахо-Корасик. Какие вершины являются терминальными в нашем автомате? (не только те, где заканчиваются строки).
37. Алгоритм Ахо-Корасик. Построение bfs-ом за $\mathcal{O}(\sum_i |s_i|)$ для произвольного алфавита.
38. Алгоритм Ахо-Корасик. Пусть есть полный автомат. Опишите алгоритм проверки наличия *каждого* словарного слова в тексте.
39. Суфф.масс. Определение. Построение стандартной сортировкой за $\mathcal{O}(n^2 \log n)$ и $\mathcal{O}(n \log^2 n)$.
40. Суфф.масс. Построение цифровой сортировкой за $\mathcal{O}(n^2)$.
41. Суфф.масс. Построение за $\mathcal{O}(n \log n)$.
42. Суфф.масс. LCP за $\mathcal{O}(n)$, алгоритм Касаи.
43. Суфф.масс. Поиск подстроки s в тексте t за $\mathcal{O}(|s| \cdot \log |t|)$.
44. Суфф.масс. Поиск общей подстроки двух строк.
45. Построение суффиксного дерева по суффиксному массиву.
46. Построение суффиксного массива по суффиксному дереву.

RMQ и LCA

47. RMQ. Формулировка задачи, решение деревом отрезков.
48. RMQ. Sparse Table.
49. RMQ. Sparse Table: сравнение с деревом отрезков.
50. RMQ. Модификации Sparse Table, решение за $[\mathcal{O}(n), \mathcal{O}(\log \log n)]$.
51. LCA. Двоичные подъёмы. `isAncestor` за $\mathcal{O}(1)$.
52. LCA. Вычисление функции на пути в дереве.
53. LCA. Эйлеров обход. Сведение LCA к RMQ. LCA за $\mathcal{O}(1)$.
54. Эйлеров обход. Функция от поддеревы за $\mathcal{O}(\log n)$.
55. LA. Формулировка. Решение задачи в *online* за $\mathcal{O}(\log n)$.
56. LA. Формулировка. Решение задачи в *offline* за $\mathcal{O}(n + m)$.