

# Вопросы к экзамену по алгоритмам

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

### Паросочетания и потоки

1. (a) Паросочетания. Формулировка задач. Сложность в произвольном и двудольном графе. Лемма о дополняющем пути.
2. (a) Паросочетания. Теорема Кёнига. Алгоритм Куна за  $\mathcal{O}(VE)$ . Версия за  $\frac{|M|}{2} \cdot \text{dfs}$ .
3. (a) Поиск контролирующего множества (вершинное покрытие, vertex cover), независимого множества (independent set), максимальной клики (max clique)  $\mathcal{O}(E)$ . Проверка графа на двудольность за  $\mathcal{O}(E)$ .
4. (b) Теорема Дилворта. Алгоритм построения максимальной антицепи. Задачи. Разбиение вершин орграфа на циклы. Разбиение массива на min число подпоследовательностей, где соседние элементы взаимно просты.
5. (b) Классификация рёбер двудольного графа – может ли, обязано ли лежать в max matching. Единственность совершенного паросочетания. Лексикографически минимальное паросочетание.
6. (a) Лемма Холла. Существование совершенного паросочетания в двудольном регулярном графе. Раскраска рёбер двудольного графа: алгоритмы за  $\mathcal{O}(k \cdot \text{Matching})$ ,  $\mathcal{O}(\log k \cdot \text{Matching})$ .
7. (b) Формулировки: теорема Брукса, теорема Визинга, теорема о 4 красках. Алгоритм для задачи Vertex-List-Coloring. Алгоритм для раскраски планарного графа в 5 цветов. Алгоритм для покраски в min число цветов, который в среднем хорошо работает.
8. (a) Количество совершенных паросочетаний. Связь с перманентом. Связь с определителем.
9. (b) Stable Matching. Алгоритм за  $\mathcal{O}(n^2)$ .
10. (a) Определение потока, величины потока, разреза, дополняющего пути, остаточной сети, циркуляции. Теорема Форда-Фалкерсона, алгоритм Форда-Фалкерсона. Целочисленность максимального потока. За сколько умеют искать max flow?
11. (a) Поиск min cut за  $\mathcal{O}(E)$ . Связь потока с паросочетанием, разреза с vertex cover. Min cost vertex cover.
12. (b) Вещественные пропускные способности, существование максимального потока. Алгоритм Эдмондса-Карпа за  $\mathcal{O}(VE^2)$ . Capacity scaling за  $\mathcal{O}(E^2 \log U)$ .
13. (b) Алгоритм Диница за  $\mathcal{O}(V^2 E)$ , скрещивание с масштабированием за  $\mathcal{O}(VE \log U)$ .
14. (c) Диниц для единичных сетей. Хопкрофт-Карпа за  $\mathcal{O}(E\sqrt{V})$ . Две теоремы Карзанова.
15. (b) Декомпозиция потока на пути и циркуляцию за  $\mathcal{O}(E^2)$ , за  $\mathcal{O}(VE)$ . Поиск  $k$ -непересекающихся путей (рёберная и вершинная версия, орграфа и неорграфа).
16. (b) Несколько истоков, стоков. Избытки и недостатки. Алгоритмы поиск LR-circulation, LR-flow, LR-max-flow за  $\mathcal{O}(\text{flow})$ .
17. (b) Preflow push. Общая схема, доказательство корректности. Простой алгоритм за  $\mathcal{O}(V^3)$ .
18. (b) Preflow push. Оптимизации: High Level, Global Relabeling. Алгоритм Ahuja за  $\mathcal{O}(VE + V^2 \log U)$ .
19. (c) Глобальный разрез. Наивный алгоритм за  $\mathcal{O}(V \cdot \text{Flow})$ . Алгоритм Штор-Вагнера. Алгоритм Каргер-Штейна за  $\mathcal{O}(V^4)$  с доказательством и за  $\mathcal{O}(V^2 \log^2 V)$  без.
20. (b) Стоимость потока. Постановка задач: min cost flow, min cost k-flow, min cost max flow, min cost circulation. Разность потоков – поток. Лемма об отсутствии циклов отрицательного веса. Поиск mincost k-flow за  $\mathcal{O}(k \cdot \text{FordBellman})$  в графе без отрицательных циклов.
21. (b) Решение задачи mincost flow. Сведение задачи mincost max flow к mincost circulation. Решение задачи mincost k-flow при наличии отрицательных циклов. Потенциалы и алгоритм Дейкстры.
22. (c) Решение задачи mincost circulation вычёркиванием циклов. Два метода. Техники cost scaling, capacity scaling. Алгоритм за  $\mathcal{O}(E \log U \cdot \text{Dijkstra})$  для поиска min cost circulation на основе capacity scaling.
23. (c) Задача о назначениях. Венгерский алгоритм.

## Строки и хеши

- a) Суффиксное дерево = ST.
  - b) Суффиксный массив = SA.
  - c) Наибольшая общая подстрока = LCS.
  - d) Наибольший общий префикс = LCP.
24. (a) Определения: суффикс, префикс, подстрока, период. Префикс функция, КМП. Z-функция, поиск подстроки в тексте. Поиск периода строки. Алгоритм Боеера-Мура.
25. (a) Полиномиальный хеш строки. Алгоритм Рабина-Карпа. Вероятность коллизии. Обработка коллизий на примере Рабина-Карпа и количества различных подстрок за  $\mathcal{O}(n^2)$ . Сравнение строк на равенство и больше-меньше. LCP за  $\mathcal{O}(\log n)$ . SA за  $\mathcal{O}(n \log^2 n)$ .
26. (a) Задача о LCS. Решение бинпоиском. Для  $k$  строк за  $\mathcal{O}(nk \log n)$ .  $\mathcal{O}(n)$  доп. памяти. Полный предподсчёт всех LCP. С помощью LCP поиск LCS за  $\mathcal{O}(n^2)$ . Поиск LCS через ST, через SA.
27. (a) Палиндромы. Дерево палиндромов за  $\mathcal{O}(n)$ . Алгоритм Манакера за  $\mathcal{O}(n)$ . Количество за  $\mathcal{O}(n \log n)$ .
28. (a) Бор. Несжатый, сжатый. 5 способов хранения. Использование: `map<string, T>`, `sort<string>`, `set<string>`. Сжатое ST, построение за  $\mathcal{O}(n^2)$ . SA + LCP  $\leftrightarrow$  ST за  $\mathcal{O}(n)$ .
29. (b) Ахо-Корасик. Ленивое построение. Построение полного автомата. Задачи. Offline поиск для каждого слова количества вхождений, самого левого и самого правого вхождения. Online поддержка суммарного количества вхождений.
30. (c) Алгоритм Укконена. Доказательство корректности и времени работы  $\mathcal{O}(n)$ .
31. (b) SA. Построение цифровой сортировкой за  $\mathcal{O}(n^2)$  и  $\mathcal{O}(n \log n)$ . Устойчивая версия. Суффиксы  $\leftrightarrow$  циклические сдвиги. Алгоритм Касаи построения LCP за  $\mathcal{O}(n)$ .
32. (c) Алгоритм Каркайнена-Сандерса построения SA за  $\mathcal{O}(n)$ .
33. (b) Полиномиальные хеши. Строка Туэ-Морса. Алгоритм построения коллизии для  $\langle P, MOD \rangle$  хеша. 3 леммы про вероятность коллизии при выборе многочлена, или точки, или обоих случайными. Как гарантировать устойчивость к взлому и отсутствие коллизий?
34. (b) Хеширование. Хеш-таблица со списками. Матожидание длин средней и максимальной цепочки. Доказательство. Двойное хеширование – утверждение про длину цепочку без доказательства. Совершенное хеширование. Алгоритмы построения хеш-функции за `time` =  $\mathcal{O}(n)$ , `space` =  $\mathcal{O}(n^2)$ ,  $\mathcal{O}(n)$ . Фильтр Блума.

## Математика

35. (a) Симметричная игра на графе. Ациклический граф. Граф с циклами. Ретроанализ. Длина игры. Длина игры в поддавки. Игра на бинарном дереве глубины  $h$ , отсечение дающее время  $\mathcal{O}(1.687^h)$ .
36. (a) Функция Гранди. Вычисление за  $\mathcal{O}(E)$ . Доказательство XOR. Выбор выигрышного хода. Примеры: задача про Карлсона, жестокая задача.
37. (a) Битовое сжатие. `bitset<N>`, основные операции. Рюкзак за  $\mathcal{O}(\frac{nS}{w})$ . Умножение и деление многочленов над  $\mathbb{F}_2$ . Умножение матриц и Гаусс над  $\mathbb{F}_2$ . Транзитивное замыкание графа за  $\mathcal{O}(\frac{n^3}{w})$  (Фloyd), за  $\mathcal{O}(\frac{nm}{w})$ .
38. (b) Четыре русских и четыре применения. Умножение матриц за  $\mathcal{O}(\frac{n^3}{w \log n})$ . Построение булевой функции из  $\mathcal{O}(\frac{2^n}{n})$  элементов. Наибольшая общая подпоследовательность за  $\mathcal{O}(\frac{n^2}{\log^2 n})$ . Дополнительное: `RMQ $\pm$ 1`.
39. (a) Гаусс над полем. Реализация за  $\mathcal{O}(nmk)$ . Треугольный и диагональный вид матрицы. Что быстрее? Решение системы линейных уравнений: количество решений, поиск решений. Линейная независимость векторов, ранг матрицы, определитель матрицы. Решение системы для  $b_1, b_2, \dots, b_t$  за  $\mathcal{O}(n(m+t)k)$ . Обратная матрица. Погрешность. Матрица Гильберта.
40. (b) Ортогонализация. Дополнение базиса до полного. Проверка, лежит ли вектор в данном подпространстве (2 способа). Расстояние от точки до подпространства (2 способа).
41. (b) Гаусс над  $Z$ . Гаусс над  $Z/mZ$  для  $m = p^k$  и произвольного. Восстановление решения.
42. (a) Марковский процесс на графе. Состояние через  $k$  итераций (2 способа). Поиск предела (2 способа). Метод итераций для решения систем уравнений.
43. (b) Равенство двух детерминированных автоматов. Эквивалентность двух детерминированных автоматов. Изоморфность двух графов.

44. **(b)** BigInteger (длинные числа). Хранение чисел (беззнаковых, знаковых, вещественных). Умножение, деление, извлечение корня за  $\mathcal{O}(\frac{n^2}{w^2})$  (правильная система счисления). Бинарный метод реализации всего. Бинарный gcd. Доказательство того, что gcd и расширенный Евклид, использующие внутри деление и умножение “за квадрат”, работают за  $\mathcal{O}(\frac{n^2}{w^2})$ . КТО, применение для BigInteger.
45. **(b)** FFT. Рекурсивный вариант. Нерекурсивный вариант. Использование для умножения чисел. Выбор системы счисления.
46. **(c)** Применение FFT. Деление и извлечение корня для чисел и многочленов за  $\mathcal{O}(n \log^2 n)$  методом “разделяй и властвуй”. Оптимизация до  $\mathcal{O}(n \log n)$  (Ньютон, удвоение точности). Перевод из одной системы счисления в другую за  $\mathcal{O}(n \log^2 n)$ .
47. **(a)** Вычисления по модулю: сложение, умножение, деление. Решето Эратосфена за  $\mathcal{O}(n \log \log n)$  и  $\mathcal{O}(n)$ . Расширенный алгоритм Евклида. Диофантовы уравнения. Решение. Класс решений.
48. **(b)** Алгоритмы проверки на простоту. Наивный. С предподсчётом простых. Тест Ферма и алгоритм за  $\mathcal{O}(n^{1/3})$ . Тест Миллера-Рабина.
49. **(b)** Факторизация. Наивный метод.  $\rho$ -эвристика Полларда. За какое время умеют решать задачу? Равносильность задач факторизации  $n = pq$  и вычисления  $\varphi(n)$ . RSA, кодирование и декодирование.
50. **(b)** Первообразный корень. Алгоритмы вычисления. Дискретный логарифм и корень  $k$ -й степени. Вычисление за  $\mathcal{O}(\sqrt{m})$ .

## 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 экзамену! 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 бор, сжатый бор?” или “за сколько вы умеете искать поток?”.
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:

- a) Меньше 1 – двойка. Ваши знания не удовлетворительны.
- b) Больше 2.5 – пятёрка. Вы круты.
- c) 2 и больше – точно четвёрка.
- d) 1.5 и меньше – точно тройка.
- e) Если от 1.5 до 2, то идёт серия простых и не очень допвопросов.