

Вопросы к экзамену по алгоритмам

СПб АУ, первый курс, весна, 2015/16 учебный год

Потоки

1. (b) Венгерский алгоритм. Задача о назначениях.
2. (a) Потоки. Def: поток, разрез, величина потока, величина разреза, остаточная сеть, циркуляция, прямые и обратные рёбра, декомпозиция потока.
3. (a) Теорема Форда-Фалкерсона. Алгоритм Форда-Фалкерсона.
4. (a) Поиск min разреза по max потоку. Декомпозиция потока за $\mathcal{O}(E^2)$, $\mathcal{O}(VE)$.
5. (a) k рёберно и вершинно непересекающихся путей в орграфе и неорграфе. Поиск паросочетания, связь контролирующего множества с разрезом.
6. (b) Алгоритм Эдмондса-Карпа. Существование max потока и целочисленного max и потока.
7. (b) Алгоритм масштабирования потока.
8. (b) Алгоритм Диница и его спаривание с масштабированием.
9. (c) Алгоритм Хопкрофта Карпа и две теоремы Карзанова.
10. (b) $[L, R]$ -циркуляция, $[L, R]$ -flow, $[L, R]$ -max-flow.
11. (c) Глобальный разрез: алгоритм Каргера-Штейна.
12. (b) Глобальный разрез: алгоритм Штор-Вагнера без доказательства. Диниц + link-cut.
13. (a) Mincost k-flow, mincost max flow, mincost circulation, транспортная задача. Формулировки задач. Критерий оптимальности mincost k-flow.
14. (a) Алгоритмы построения mincost k-flow и mincost flow через дополняющие пути в графе без отрицательных циклов.
15. (a) Графы с отрицательными циклами. Алгоритм Клейна. Полиномиальная версия.
16. (b) Mincost k-flow. Дейкстра и потнециалы.
17. (c) Capacity scaling. Псевдополиномиальный алгоритм для mincost k-flow.

Строки и хеширование

18. (a) Префикс функция. Поиск подстроки в строке и периода строки.
19. (a) Z-функция. Поиск подстроки в строке и периода строки.
20. (b) Алгоритмы Бозра-Мура поиска подстроки в строке.
21. (a) Хеши: полиномиальный хеш, хеш подстроки, Алгоритм Рабина-Карпа поиска подстроки в строке с $\mathcal{O}(1)$ допамяти.
22. (b) Хеши: вероятности. 3 леммы. Каким хешом пользоваться? Обоснование вероятности коллизии в худшем и в среднем.
23. (c) Хеши: антихеш тесты. Строка Туэ-Морса, тест против (p, mod) -хеша.
24. (a) Хеши: поиск общей подстроки, поиск LCP, построение суффмассива за $\mathcal{O}(n \log^2 n)$.
25. (a) Суффмассив: построение за $\mathcal{O}(n \log n)$. Поиск подстроки в тексте.
26. (a) Суффмассив: алгоритм Касаи за $\mathcal{O}(n)$.
27. (b) Суффмассив: построение за $\mathcal{O}(n)$, Каркайнен-Сандерс.
28. (a) Бор, сжатый бор, хранение. Построение суффдерева за $\mathcal{O}(n^2)$ двумя способами. Подстрока в тексте.
29. (b) Автомат Ахо-Корасик. Поиск словарных слов в тексте.
30. (c) Суффдерево: алгоритм Укконена построения за $\mathcal{O}(n)$.
31. (a) Преобразование суффдерева \leftrightarrow (суффмассива + LCP) за $\mathcal{O}(n)$.
32. (a) Решение задач суффмассивом и суффдеревом: поиск общей подстроки, число различных подстрок.
33. (b) Универсальное семейство хеш-функций. Пример семейства. Применение: хеш-таблица на списках.
34. (b) Совершенное хеширование: двухуровневая схема, схема на основе ациклического графа.
35. (b) Хеширование кукушки. Фильтр Блюма (рассказан в конце).

Математика

36. (b) Симметричные игры на графах. Ациклический граф. Граф с циклами. Прямая сумма игр и функция Гранди.
37. (a) Расширенный алгоритм Евклида, обращение числа, поиск обратных для чисел $1..k$ за $\mathcal{O}(k)$. Обращение через возведение в степень.
38. (a) RSA. Схемы с симметричным и открытым ключом. Алгоритм кодирования, декодирования, сложность вычислений.
39. (a) Первообразный корень. Алгоритм за $\mathcal{O}(\text{FACT})$, вероятностный за $\mathcal{O}(\log \log n)$.
40. (b) Фурье. Схема для умножения многочленов. Прямое преобразование. Обратное преобразование без доказательства.
41. (b) Фурье. Обратное преобразование: доказательство. Использование для умножения чисел, выбор системы счисления, оценка погрешности.
42. (b) Фурье. Предподсчёт корней. Нерекурсивная реализация, разворот битовой записи числа.
43. (b) Длинная арифметика: хранение, линейные операции, деление и умножение многочленов за квадрат, деление чисел за $\mathcal{O}(\frac{n^2}{w^2})$.
44. (a) Двоичная арифметика: умножение, деление, gcd.
45. (c) Деление чисел и многочленов за $\mathcal{O}(n \log^2 n)$ методом разделяй и властвуй.
46. (b) Сложность. NP, NPC, NPh. Сведения по Карпу и Куку. Примеры задач из разных классов.
47. (b) Примеры сведений: CNF-SAT \rightarrow 3-SAT \rightarrow IS. Эквивалентность decision и search для NP.
48. (a) Гаусс. Приведение матрицы к трапецевидному виду и подсчёт определителя за $\mathcal{O}(n^3)$. Разложение вектора в базисе и поиск обратной матрицы за $\mathcal{O}(n^3)$. Гаусс над \mathbb{F}_2 за $\mathcal{O}(n^3/w)$.
49. (a) Гаусс. Над областью целостности. Поддержание базиса линейной оболочки вектора: добавление вектора за $\mathcal{O}(nk)$.
50. (a) Метод итераций. КТО, использование КТО для вычислений в длинных числах.
51. (a) Решето Эратосфена за $\mathcal{O}(n)$.
52. (b) Проверка на простоту: алгоритм за $\mathcal{O}(n^{1/2}/\log n)$, вероятностный $\mathcal{O}(n^{1/3}/\log n)$, вероятностный $\mathcal{O}(\log n)$.
53. (b) Факторизация: ρ -эвристика Полларда за $\mathcal{O}(n^{1/4}\text{gcd})$. Обоснование в предположении равномерного распределения.
54. (b) Дискретный логарифм, извлечение корня k -й степени по модулю.
55. (b) Вероятностные классы. Randomized, probabilistic. RP, coRP, ZPP. Теорема $\text{ZPP} = \text{RP} \cap \text{coRP}$.
56. (b) Вероятностные классы. Классификация ошибок: отсутствует, односторонняя, двусторонняя. BPP, понижение вероятности. Пример задачи 3-List-Coloring, показывающий, что не любую вероятность можно понизить, оставаясь в BPP.

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 хотя бы помнить, что вы должны знать, и 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 LCA?” или “за сколько вы умеете считать минимум на пути дерева?”.
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 – минуса к оценке: $(c) = -0.2$, $(b) = -0.4$, $(a) = -0.6$.
7. Заменять билет можно только один раз.
8. 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, а какие нет, 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 1.33 – 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.