Вопросы к экзамену по доп главам алгоритмов
SPb HSE, ВШЭ 3-й курс, 18 июня 2024

Общая информация

- Конспект: [pdf]
- Экзамен устный. ≈120 минут на ответ.
  Экзамен проходит *без* использования конспекта и других источников.
- В каждой теме есть теорминимум, который обязательно нужно знать. Кроме того экзаменуемый самостоятельно выбирает 40% вопросов, и готовит их к сдаче. На экзамене нужно поставить плюсики напротив вопросов, котороые вы готовы отвечать. 12 вопросов – обязательная база; 10 вопросов – дополнение; из оставшихся 96 вопросов нужно $\boxed{40\% = 38 \text{ вопросов.}}$
- Если не знать теорминимум, оценка неудовлетворительная.
  В остальном оценка линейна от 4/10 до 10/10.
- (*) – теорминимум.
- (a) – алгоритмы.
- (b) – доказательства.
- (c) – особо сложные темы.
- (+) – факультативные темы для тех, кому хочется.

Многочлены, FFT и применения

(*) 1. Что такое FFT? За сколько работает? Опишите любую рабочую версию прямого FFT.
Как с помощью FFT посчитать число счастливых билетов?

(a) 2. Базовые действия над F_2: умножение за $\mathcal{O}(\frac{nm}{w})$, деление за $\mathcal{O}(\frac{(n-m)m}{w})$, gcd за $\mathcal{O}(\frac{nm}{w})$.

(a) 3. Быстрое умножение над F_2: $\mathcal{O}(nm/w^{1.58})$. Быстрое умножение над кольцом.

(a) 4. Рекурсивное прямое FFT, обратное FFT, нерекурсивное FFT, предподсчёт корней.

(b) 5. Обратное FFT. Точность вычислений. FFT по модулю. FFT над $\mathbb{Z}$.

(a) 6. Разделяй и властвуй: смена системы счисления; деление; значения в точках; интерполяция.

(a) 7. Деление многочленов за $\mathcal{O}(n \log n)$. Деление чисел за $\mathcal{O}(n \log n)$.

(a) 8. *Задачи:* 3-SUM; $n!$ за $\mathcal{O}^*(\sqrt{n})$; факторизация за $\mathcal{O}^*(n^{1/4})$.

(a) 9. *Задачи:* FFT(x, y); покраска в k цветов; рюкзак «набрать S ровно k предметами» за $\mathcal{O}^*(nS)$.

(a) 10. Рюкзак за $\mathcal{O}^*(S\sqrt{n})$.

(+) 11. Рюкзак за $\mathcal{O}^*(S + n)$.

(a) 12. CRC-32, вычисление за $2n$ битовых операций.
Кодирование с ошибками: проверка одной ошибки; исправление одной ошибки.

(b) 13. Какие бывают конечные поля? Как получить поле $F_{2^{32}}$?

(a) 14. Коды Рида-Соломона (кодирование для исправления k ошибок). Кодирование. Проверка наличия ошибок. Декодирование при отсутствии ошибок, при наличии ошибок. Время работы.

(b) 15. Критерий корректности кода через расстояние Хэмминга. Корректность кодирования.
Корректность декодирования. Почему СЛАУ имеет единственное решение?

(a) 16. Линейная рекуррента за $\mathcal{O}(k^3 \log n)$ и $\mathcal{O}(k \log k \log n)$.

Автоматы и суффиксные автоматы

(*) 17. Что такое детерминированный (DA) и недетерминированный (NDA) автоматы? Что такое суффиксный автомат? Правый контекст? Суффиксная ссылка? Какие конечные состояния у суффавтомата? Как с помощью суффавтомата искать строку s в тексте t за $\mathcal{O}(s)$?

(a) 18. Принятие строки DA за $\mathcal{O}(s)$ и NDA за $\mathcal{O}(m \cdot |s|)$, детерминизация автомата.

(a) 19. Проверка эквивалентности двух автоматов за $\mathcal{O}(m^2)$.

(a) 20. Минимизация Хопкрофтом за $\mathcal{O}(nk \log n)$.

(a) 21. Изоморфность автоматов за $\mathcal{O}(m)$, эквивалентность минимальному за $\mathcal{O}(m)$, минимизация недетерминированного: алгоритм Бржозовского.

(b) 22. Минимизация: правые контексты, разбиение вершин на классы эквивалентности, рёбра, корректность определения, определение суффиксного автомата, суффссылок, терминальные вершины суффавтомата. Леммы, связанные с правыми контекстами.

(a) 23. Алгоритм построения суффиксного автомата за линейное время.

(b) 24. Корректность алгоритма (почему раздвоится только одна вершина, и мы именно её находим).

(b) 25. Оценка $\mathcal{O}(n)$ на время построения.

(b) 26. Размер: оценки $2n$ вершин, $3n$ рёбер. Суффдерево по суффавтомату, обоснование.

(a) 27. *Задачи:* LZSS за $\mathcal{O}(n)$; общая подстрока k строк с доппамятью $\mathcal{O}(\min |s_i|)$; refrain.

SAT-ы

(*) 28. Определения: NP, coNP, SAT, UNSAT, TAUT.
Что такое доказательство? Доказательство длины 2^n для SAT.

(a) 29. Алгоритм DPLL для решения SAT. Unit Clause Elimination.

(b) 30. Методы доказательств: resolutions, tree-like-resolutions.
DPLL для поиска доказательства неразрешимости SAT методом резолюций.

(c) 31. Алгоритм CDCL для решения/доказательства SAT. Эвристики выбора литерала.

Паросочетания в произвольном графе

(*) 32. Алгоритм поиска паросочетаний в произовольном графе за $\mathcal{O}(V^2E)$ без подробности реализации и без доказательства.

(b) 33. Лемма о дополняющем пути. Лемма имени Куна.

(a) 34. Алгоритм поиска паросочетания сжатием соцветий за $\mathcal{O}(V^3)$.

(b) 35. Обоснования: корректность алгоритма сжатия соцветий

(a) 36. Реализация Габова алгоритма сжатия соцветий за $\mathcal{O}(V^3)$ без рекурсивного сжатия.

(b) 37. Обоснования: корректность реализации Габова (почему ссылки `p[]` не циклятся?).

(a) 38. Оптимизации: алгоритм за $\mathcal{O}(|M| \cdot V^2)$, `dfs` vs `bfs`, поиск LCA за $\mathcal{O}(1)$.

(a) 39. Оптимизации: жадная инициализация; DSU; $\mathcal{O}(|M| \cdot E \cdot \alpha)$.

(a) 40. *Задача:* поиск минимального по весу простого пути из чётного числа рёбер.

Линейное программирование и симплекс метод

(*) 41. Сформулируйте задачи LP и ILP. Запишите задачу mincost-k-flow в форме LP. Определите двойственную задачу, сформулируйте теорему сильной двойственности. Что такое полиэдр? Сформулируйте кратко симплекс-метод на языке «вершины и рёбра полиэдра».

(b) 42. LP и ILP задачи: кратчайшее расстояния, flow, multicommodity-flow, два непересекающихся пути, mincost-max-matching (двудольный граф, произвольный граф), mincost-vertex-cover.

(b) 43. Полиэдр. Выпуклость. Почему оптимум в вершине? Почему ненулевых x_i мало?
Алгоритм за $\mathcal{O}(2^{n+m}n^3)$ перебором базисных планов. Корректность.

(a) 44. Симплекс метод, если уже дано начальное решение. Корректность Симплекс метода.

(a) 45. Поиск начального решения для Симплекс метода. Корректность.

(a) 46. Решение задачи LP для малых размерностей d за $\mathcal{O}(n \cdot d!)$.

(a) 47. Обучение Перцептрона. Оценка времени работы.

(b) 48. Слабая теорема двойственности. Доказательство.

(b) 49. Сильная теорема двойственности. Доказательство.

(b) 50. TUM. Определение. TUM матрицы инцидентности двудольного. Проблемы с недвудольным. Теорема: TUM $\Rightarrow$ Симплекс даст целочисленное решение.

(a) 51. Метод Эллипсоидов. Постановка задачи. Алгоритм. Время работы без обоснования.

(b) 52. Матричные игры. Вероятностная стратегия. Нэш: существование равновесия вероятностных стратегий для симметричных игр для двух игроков. Поиск вероятностной стратегии.

(+) 53. Метод внутренней точки.

(+) 54. SDP. Постановка задачи. 0.878-OPT решение для MAX-CUT.

(+) 55. Выпуклая оболочка в d-мерном пространстве. Пример из $2d$ вершин и 2^d граней. Нахождение множества вершин выпуклой оболочки симплексом.

Факторизация

(*) 56. Сформулируйте задачу «факторизация». Как gcd помогает решить задачу «факторизация»? За сколько работает метод Крайчика?

(a) 57. Крайчик, проверяющий m чисел на b-гладкость за $\mathcal{O}(mb)$, и работающий за $\mathcal{O}(mb+b^3)$.

(a) 58. Крайчик, проверяющий m чисел на b-гладкость за $\mathcal{O}(m \log\log m)$, и работающий за $\mathcal{O}(m \log\log m + b^2 \log m)$. Экономная по памяти реализация Крайчика.

(a) 59. Извлечение корня по простому модулю. $\mathcal{O}(p^{1/2})$. $\mathcal{O}(\log p)$.

(+) 60. Обоснование времени работы.

Алгоритмы на графах

(*) 61. Определения: трёхсвязность; ориентированное MST; доминатор, дерево доминаторов. Задача динамической связности. За сколько решается в online? (для неорграфа, для орграфа).

(a) 62. Выделение компонент трёх-связности за $\mathcal{O}(n)$.

(a) 63. Алгоритм Эппштейна поиска k-го пути за $\mathcal{O}((n+m+k)\log n)$.

(a) 64. Ориентированное MST. Алгоритм двух китайцев за $\mathcal{O}(m \log n)$ без восстановления ответа.

(a) 65. Алгоритм двух китайцев: восстановления ответа. Способ Примом, способ за линейное время.

(a) 66. Динамические орграфы: добавление рёбер за $\mathcal{O}(\frac{n}{w})$; full-dynamic-DAG за $\mathcal{O}(n^2)$.

(a) 67. Динамические орграфы: удаление рёбер в online (Shiloach'81) за $\mathcal{O}(q+nm)$.

(a) 68. Динамическая связность в offline за $\mathcal{O}(m \log^2 m)$ с DSU.
Задача: точки на плоскости добавляются, удаляются, поддерживать 2 ближайшие.

(a) 69. Динамическая связность в offline за $\mathcal{O}(m \log m)$; динамическое MST за $\mathcal{O}(m \log m)$.

(a) 70. Динамическая связность в online: базовые решения, включая ETT; перебор в ETT вершин с положительным количеством интересных рёбер за $\mathcal{O}(k \log n)$.

(c) 71. Динамическая связность в online для неорграфа.

(a) 72. Дерево доминаторов: определение; простейшие алгоритмы построения;
построение в ацикличном графе за $\mathcal{O}(m\alpha)$; lowerbound на время работы через LCA.

(+) 73. Дерево доминаторов: $\mathcal{O}(m\alpha)$ в общем случае с обоснованием.

Планарные графы

(*) 74. Что такое планарный граф, плоский граф? Сформулируйте теорему Эйлера, теорему Куратовского, теорему Липтона-Тарьяна про сепаратор.

(b) 75. Теоремы: $E \le 3V - 6$, $E \le 2V - 4$ с доказательством.
Доказательство теоремы Куратовского в одну из сторон.

(a) 76. Алгоритм Демукрона проверки на планарность. Оценка времени $\mathcal{O}(n^2)$.

(a) 77. Пружинная теорема. Укладка графа прямыми отрезками.

(b) 78. Доказательство того, что для СЛАУ из пружинной теоремы $\exists!$ решение.

- (b) 79. Трёхсвязность. Как сделать планарный граф трёхсвязным (с доказательством)?
  Что ломается при отсутствии трёхсвязности?
- (a) 80. Алгоритм поиска сепаратора. MAX-IS и к-COLORING за $2^{\Theta(\sqrt{n})}$.
- (b) 81. Теорема Липтона-Тарьяна о существовании сепаратора с доказательством.
- (a) 82. Выделение граней плоского графа за $\mathcal{O}(sort(n))$.
- (a) 83. Gauss Elimination за $\mathcal{O}(n^{3/2})$. Закон Кирхгоффа (сопротивление сети проводников).

Матроиды

- (*) 84. Что такое матроид? Что такое база матроида? Пример: графовый матроид.
  Сформулируйте алгоритм Радо-Эдмондса поиска базы максимального/минимального веса.
- (b) 85. Примеры матроидов с обоснованием «почему матроид».
  Универсальный (две версии). Графовый. Матричный (векторный). Трансверсальный.
- (b) 86. Алгоритм Радо-Эдмондса с доказательством. Применение для матроидов выше.
- (b) 87. Определение пересечения матроидов, объединения матроидов,
  сведение задачи объединения k матроидов к задаче пересечения двух.
- (b) 88. Th: объединение – всегда матроид. Доказательство. Пример того, что пересечение –
  не обязательно матроид. Доказательство того, что пересечение трёх матроидов $\in$ NP-н.
- (a) 89. Алгоритм поиска пересечения матроидов. Оценка времени работы.
- (b) 90. Обоснование алгоритма пересечения матроидов: почему, если дополняющего пути в графе замен нет, то найдено максимальное пересечение?
- (+) 91. Обоснование алгоритма пересечения матроидов: $I \triangle P \in \mathcal{I}$, если P – кратчайший.
- (a) 92. Задачи, решение с помощью матроидов, оценка времени работы: поиск разноцветного дерева; поиск k непересекающихся остовов; покрытие множества рёбер минимальным числом остовов.

Структуры данных

- (*) 93. X-fast-trie. Всё о структуре (что она умеет, за сколько, как именно).
- (a) 94. Поиск mex на отрезке в online. Статический массив и $\mathcal{O}(\log n)$; изменение в точке и $\mathcal{O}(\log^2 n)$.
- (a) 95. LCA: двоичные подъёмы с $\mathcal{O}(n)$ памяти. RMQ: решение за $[\mathcal{O}(n), \mathcal{O}(1)]$
- (+) 96. Разбиение отрезков. Замыкание Disjoint-Sparse-Table до $[n \cdot \alpha, \alpha]$.
- (a) 97. X-fast-trie, Y-fast-trie, сравнение с V.E.B.
- (c) 98. Fusion tree. Получение BST за $\mathcal{O}(n \log_w n)$ без описания математического оператора $sketch(x_i)$.
  Вывод сортировки за $\mathcal{O}(n\sqrt{\log n})$.

Суффиксный массив для биоинформатиков

- (*) 99. Что такое BWT, BWT^{-1}? За какое время их умеют строить?
  Какую задачу решает fm-index, чем он лучше бинпоиска по суффиксному массиву?
- (a) 100. Сортировка строка за $\mathcal{O}(L + |A|)$, где L – сумма длин, A – алфавит.
- (a) 101. Алгоритм SA-IS построения суффиксного массива.
- (a) 102. BWT^{-1} за $\mathcal{O}(n)$.
- (a) 103. fm-index. Построение за $\mathcal{O}(n)$, ответ на запрос за $\mathcal{O}(|s|)$.

Вычислительная геометрия

(*)104. Определения: диаграмма Вороного, триангуляция Делоне.
Алгоритм построения диаграммы Вороного за $\mathcal{O}(n^2)$.

(a)105. Локализация точки на плоскости. Формулировка задачи, решение.

(a)106. Построение выпуклой оболочки: Грэхем $\mathcal{O}(n\log n)$, Эндрю $\mathcal{O}(sort(n))$, Джарвис $\mathcal{O}(nk)$,
Чен $\mathcal{O}(n\log k)$, если нам подсказали верное k.

(a)107. Выбор k для Чена, улучшение времени работы Чена до $\mathcal{O}(\frac{n}{k}\cdot sort(k))$.

(a)108. Биекция прямых и точек. Пересечение полуплоскостей за $\mathcal{O}(n\log k)$.

(a)109. Рандомизированные алгоритмы. Две ближайших точки за $\mathcal{O}(n)$.
Минимальный покрывающий круг за $\mathcal{O}(n)$. Точка пересечения n кругов за $\mathcal{O}(n)$.

(a)110. Задачи. Выбрать k точек из n: диаметр $\to\min$. Перечислить все пары точек: расстояние $\le d$.

(a)111. Построение триангуляции Делоне за $\mathcal{O}(n\log n)$. Только алгоритм.

(b)112. Построение триангуляции Делоне за $\mathcal{O}(n\log n)$. Обоснование времени работы.

(a)113. Сумма Минковского, определение, построение за $\mathcal{O}(n+m)$, использование для поиска пересечения движущихся выпуклых многоугольников.

(a)114. Motion Planning. Случай невыпуклых препеятствий. Применение суммы Минковского.
Граф видимости. Поиск кратчайшего пути за $\mathcal{O}(n^3)$.

(c)115. Motion Planning. Поиск кратчайшего пути за $\mathcal{O}(n^2\log n)$, произвольного пути за $\mathcal{O}(n\log^2 n)$.

(a)116. Динамическая выпуклая оболочка. $\mathcal{O}(\log^2 n)$ на запрос.
Применения триангуляций: для каждой пары найти ближайшую точку; MST на плоскости.

(+)117. Число точек в полуплоскости онлайн за $o(n^{0.5})$.

(+)118. Даны n точек на плоскости. 3D-ортогональный-запрос минимум за $\mathcal{O}(\log n)$.