

Список вопросов к экзамену по курсу «Алгоритмы и структуры данных» (второй модуль)

Графы

1. Поиск в ширину (BFS), доказательство корректности.
2. Алгоритм Дейкстры, доказательство корректности. Реализации для плотных и разреженных графов.
3. Алгоритм Форда—Беллмана. Оптимизация памяти до линейной, доказательство корректности.
4. Алгоритм Флойда—Уоршелла. Оптимизация памяти до квадратичной, доказательство корректности.
5. MST: формулировка задачи. Лемма о безопасном ребре (свойство разреза).
 - (a) Алгоритм Прима, реализации за $\mathcal{O}(V^2)$, $\mathcal{O}(E \log V)$.
 - (b) Алгоритм Краскала. Система непересекающихся множеств: лес с двумя эвристиками без доказательства оценки времени работы.
6. Реализации СНМ: массив, сливаемые вектора, лес, лес с эвристикой рангов. Оценки времени работы.
- (c) 7. Реализация СНМ: лес с эвристиками рангов и сжатия путей. Оценка амортизированного времени работы $\mathcal{O}(\log^* n)$.
8. Потоки: мотивация, транспортная сеть, поток и его величина, остаточная сеть, разрез и его величина.
 - (a) Теорема (вместе с необходимыми леммами) и алгоритм Форда—Фалкерсона.
 - (b) Целочисленная сеть, на которой алгоритм Форда—Фалкерсона работает экспоненциальное от длины входа время (порядок обхода рёбер не фиксирован). Алгоритм Форда—Фалкерсона с масштабированием. Алгоритм Эдмондса—Карпа (без доказательства оценки времени работы), существование максимального потока.

Бинарные деревья поиска (BST)

9. BST как SortedSet и SortedMap: поиск, вставка, удаление, центрированный (inorder) обход, поиск следующего и предыдущего элемента, lower_bound, k -я порядковая статистика. AVL-дерево. Оценка $\mathcal{O}(\log n)$ на высоту.
- (c) 10. Большие и малые вращения. Перебалансировка. Псевдокод операции добавления в AVL-дерево.

Алгоритмы на строках

11. Задача поиска подстроки в строке. Наивный алгоритм.
 - (a) Z-функция и её применение.
 - (b) Префикс-функция, алгоритм Кнута—Морриса—Пратта.
 - (c) Полиномиальное хеширование, алгоритм Рабина—Карпа. Оптимизация памяти до константной.
- (c) 12. Оценки вероятности коллизии при полиномиальном хешировании: k сравнений случайных строк, k сравнений произвольных строк.

13. Хеш-таблица: с закрытой, с открытой адресацией. Универсальное семейство хеш-функций. Время работы хеш-таблицы с закрытой адресацией. Полиномиальное хеширование и хеш-таблица: композиция полиномиального и универсального хеширования.
14. Задача поиска словарных слов в тексте. Бор.
Решение за $\mathcal{O}(|\Sigma| \cdot \sum_i |\text{pattern}_i| + |\text{text}| \cdot \max_i |\text{pattern}_i|)$.
- (с) 15. Суффиксные ссылки и переходы автомата Ахо—Корасик на боре. Подсчёт за линейное от размера бора время. Решение задачи поиска словарных слов в тексте за $\mathcal{O}(|\Sigma| \sum_i |\text{pattern}_i| + |\text{text}|)$.

Числовые алгоритмы

16. Модульная арифметика: отношение сравнимости, свойства, деление и обратный элемент по модулю (условие существования, без нахождения). Алгоритм Евклида, оценка количества арифметических операций. Расширенный алгоритм Евклида. Линейное диофантово уравнение.
17. Обратный элемент по простому модулю: малая теорема Ферма, быстрое возведение в степень. Обратный элемент по произвольному модулю: расширенный алгоритм Евклида.
18. Разложение на множители и проверка на простоту — обзор. Решето Эратосфена. Оценка времени работы, доказательство оценки $\mathcal{O}(n \log n)$.
19. Асимметричное шифрование: мотивация, общая схема. Алгоритм RSA.
20. Китайская теорема об остатках. Использование для длинной арифметики.

NP-трудные задачи

21. Задачи поиска и распознавания.
 - (а) Класс **NP** задач *поиска*, примеры таких задач (с доказательством). Связь между задачами максимизации/минимизации и задачами поиска.
 - (б) Классы **NP** и **P** задач *распознавания*, примеры таких задач (с доказательством). $\mathbf{P} \subseteq \mathbf{NP}$.
22. Полиномиальные сведения и их свойства: транзитивность, замкнутость **P** и **NP**. **NP**-трудность (**NPH**), **NP**-полнота (**NPC**). Свойства **NP**-трудных задач.
23. **NP**-полнота задачи **CIRCUIT SAT** (нестрогое доказательство). Теорема Кука—Левина: $\text{SAT} \in \text{NPC}$ [на практике]. Примеры пар похожих задач, одна из которых лежит в **P**, а другая является **NP**-полной (без доказательств).

Формат проведения экзамена

Сначала все пишут тест. Тест определяет вашу максимальную оценку за экзамен (см. ниже).

Затем вы подключаетесь в выделенное вам время (ближе к экзамену появится таблица для записи), между тестом и временем подключения у вас перерыв. При подключении вам выдаётся *несколько* вопросов из списка и **7 минут готовитесь**. Таким образом, время на подготовку — это время на то, чтобы вспомнить, о чём выданные вопросы и, возможно, подготовку глобального плана рассказа.

Перед выдачей вопросов вы можете выбрать упрощённый режим. В таком режиме вы точно не получите вопросы, помеченные (с), но ваша оценка за устный ответ будет умножаться на 0.7 (то есть полный ответ даёт 7 баллов, половина ответа — 3.5 балла).

Пусть s_t — оценка за тест, а s_e — оценка за устный ответ. Тогда оценка за экзамен — $\min(s_t, s_e)$.

В течение всего экзамена (как теста, так и устного ответа) пользоваться дополнительными материалами (конспектами, книгами, интернетом и т. д.) не разрешается.

В течение всего экзамена (как теста, так и устного ответа) может потребоваться включённая камера (на которой видно вас) и демонстрация экрана (не отдельного окна, но всего экрана).

Во время ответа вам наверняка нужно будет что-нибудь писать так, чтобы это видел экзаменатор. Например, можно демонстрировать графический и/или текстовый редактор на своём компьютере (это, в том числе, может зависеть от характера вопроса — в некоторых удобнее рисовать, в некоторых писать).

Пожалуйста, подумайте заранее об том, как лучше это организовать. Постарайтесь сделать так, чтобы скорость вашего рассказа была не сильно ниже по сравнению с тем, как если бы вы отвечали очно. (Например, если вы пишете текст мышкой в графическом редакторе, это часто оказывается сильно медленнее.)