

Правила проведения экзамена по курсу
«Алгоритмы и структуры данных»
Осенний семестр, 2022-23

Экзамен будет проходить в четыре этапа: три основных и один бонусный.

1. Выдается билет, содержащий четыре вопроса из программы: по два вопроса из первой и второй категорий, при этом на тематику каждого из контестов не более одного вопроса. На подготовку ответа дается один час. За ответ на эту часть программы дается до шести баллов. Продолжительность ответа в ходе данной части экзамена не должна превышать одного часа.

Экзаменатор имеет право попросить сформулировать смежные по теме определения и формулировки теорем, не указанные явно в вопросе. При использовании теоремы из данного курса экзаменатор имеет право поинтересоваться ее доказательством.

2. Далее идет блиц-опрос по программе, за который дается не более одного балла. В его ходе экзаменатор имеет право спросить любое определение и формулировку любой теоремы из курса. Также он может поинтересоваться постановкой задачи, которую решает некоторый алгоритм. Отвечать необходимо практически сразу, без долгих размышлений.
3. Заключительная часть экзамена состоит в том, что вам будут предложены для решения две задачи: одна попроще и одна посложнее. Для получения более сложной задачи необходимо решить сначала задачу попроще. Каждая задача принесет вам еще один балл. Продолжительность времени решения простой задачи не превышает тридцати минут, более сложной — сорока пяти.
4. В случае, если вы претендуете на отл(10), вам нужно сообщить об этом экзаменатору. После консультации с лектором он выдаст либо вопрос со звездочкой из программы, либо же задачу продвинутого уровня сложности. Подготовка и ответ этого этапа не должны превышать полутора часов.

В случае, если на какой-то из стадий экзамена истекло выделенное время, экзаменатор имеет право прервать экзамен и выставить оценку на основе данных вами ответов.

Если вам необходимо отлучиться в ходе экзамена, это можно сделать только между стадиями или между задачами из третьего этапа.

В случае несогласия с оценкой вы имеете право оапеллировать ее у лектора до выхода из аудитории.

Программа к экзамену по курсу «Алгоритмы и структуры данных»
Осенний семестр, 2022-23

1. (1) Понятие временной и пространственной (по памяти) сложности алгоритма. Определение асимптотических сравнений \mathcal{O} , Θ , Ω . Примеры.
2. (1) Бинарный поиск. Бинарный поиск по ответу. Временная сложность.
3. (1) Вещественнозначный бинарный поиск (по ответу). Временная сложность.
4. (1) Префиксные суммы. Обобщение на произвольную обратимую операцию.
5. (1) Линейные контейнеры. Односвязный и двусвязный списки.
6. (1) Линейные контейнеры. Стек, очередь, дек (через списки и через кольцевой буфер).
7. (2) Линейные контейнеры. Очередь с поддержкой произвольной ассоциативной операции.
8. (1) Амортизированный анализ. Метод монеток. Применение на примере `push_back` в динамическом массиве.
9. (2) Амортизированный анализ. Метод потенциалов. Применение на примере `push_back` в динамическом массиве.
10. (1) Бинарная пирамида. Определение. Поддержка свойства пирамиды.
11. (1) Бинарная пирамида. Определение. Основные операции.
12. (2) Бинарная пирамида. Определение. Операция `Heapify`. Пирамидальная сортировка.
13. (1) Сортировки. Теорема о нижней границе времени сортировки.
14. (1) Сортировки. Стабильные сортировки. Сортировка слиянием.
15. (*) Теорема о нижней границе времени `Merge` (формула Стирлинга б/д).
16. (1) Быстрая сортировка. Оценка времени (б/д).
17. (2) Алгоритм медианы медиан. Оценка времени работы быстрой сортировки с его использованием.
18. (1) Поиск порядковой статистики. Оценка времени (б/д).
19. (2) Алгоритм медианы медиан. Оценка времени поиска порядковой статистики с его использованием.
20. (1) Поразрядные сортировки. LSD.
21. (1) Бинарное дерево поиска. Определение, основные операции (без их алгоритма). Обходы деревьев. Применение.
22. (1) Бинарное дерево поиска. Определение, основные операции.
23. (1) Бинарное дерево поиска. Определение. Поиск порядковой статистики, `lower_bound`.
24. (1) AVL-дерево. Определение. Теорема о высоте.
25. (2) AVL-дерево. Определение. Балансировка. Основные операции.

26. (2) Декартово дерево. Определение. Операции **split**, **merge**. Выразимость остальных операций через них. Теорема о высоте $\mathcal{O}(\log n)$.
27. (2) Splay-дерево. Определение. Операция **splay**. Выразимость остальных операций через них. Теорема о высоте $\mathcal{O}(\log n)$.
28. (1) В-дерево. Определение. Теорема о высоте. Процедура поиска.
29. (2) В-дерево. Определение. Теорема о высоте ($\mathcal{O}(\log n)$). Процедура вставки.
30. (*) В-дерево. Определение. Теорема о высоте ($\mathcal{O}(\log n)$). Процедура удаления.
31. (1) Задачи **RMQ**, **RSQ**. Постановки **online/offline**, **static/dynamic**.
32. (1) Разреженная таблица. Область применимости. Построение. Ответ на запрос.
33. (1) Дерево отрезков. Область применимости. Построение. Изменение в точке. Ответ на запрос в случае отсутствия группового обновления.
34. (2) Дерево отрезков. Область применимости. Построение. Изменение в точке. Ответ на запрос в случае наличия группового обновления.
35. (1) Дерево Фенвика. Область применимости. Построение за линейное время. Изменение в точке. Ответ на запрос. Теорема о связи функций f и g $\mathcal{O}(\log n)$.
36. (2) Теорема о связи функций f и g в дереве Фенвика.
37. (2) Декартово дерево по неявному ключу. Задача о развороте подотрезка.
38. (1) Хеш-функция. Коллизия. Хеш-таблица и основные операции. Хеш-таблица на цепочках (время работы $\mathcal{O}(\log n)$).
39. (1) Гипотеза простого равномерного хеширования. Математическое ожидание длины цепочки.
40. (2) Универсальное семейство хеш-функций. Теорема о сильной универсальности $\mathcal{H} = \{h_{a,b}\}_{a \in \mathbb{Z}_p^*, b \in \mathbb{Z}_p}$, где $h_{a,b}(x) = ((ax + b) \% p) \% m$.
41. (1) Задача о наибольшей возрастающей подпоследовательности. Решение за квадратичное время.
42. (2) Задача о наибольшей возрастающей подпоследовательности. Решение за $\mathcal{O}(N \log N)$, где N — длина последовательности.
43. (1) Задача о наибольшей общей подпоследовательности.
44. (1) Задача о 0-1 рюкзаке.
45. (1) Матричное динамическое программирование. Подсчет количества *гладких* чисел длины k за $\mathcal{O}(k)$.
46. (1) Матричное динамическое программирование. Подсчет k -го члена однородной линейной рекурренты n -го порядка за $\mathcal{O}(n^3 \log k)$.
47. (2) Матричное динамическое программирование. Подсчет k -го члена неоднородной линейной рекурренты n -го порядка с полиномиальной неоднородностью порядка t за $\mathcal{O}((n + t)^3 \log k)$.
48. (2) Матричное динамическое программирование. Подсчет количества *гладких* чисел длины k за $\mathcal{O}(\log k)$.

- 49. (1) Динамическое программирование на подотрезках. Задача об оптимальном перемножении матриц.
- 50. (1) Динамическое программирование на подмножествах. Задача коммивояжера с допущением кратного посещения одного населенного пункта.
- 51. (1) Геометрия. Прямая, отрезок. Взаимное расположение.
- 52. (1) Геометрия. Многоугольники. Проверка на выпуклость. Площадь многоугольника (любой способ на выбор).
- 53. (1) Геометрия. Многоугольники. Принадлежность точки. Общий случай и множество запросов для выпуклого.
- 54. (1) Выпуклые оболочки. Алгоритм Джарвиса.
- 55. (1) Выпуклые оболочки. Алгоритм Грехема.
- 56. (2) Сумма Минковского. Теорема о форме суммы Минковского двух выпуклых многоугольников (выпуклый многоугольник). Алгоритм построения для двух выпуклых многоугольников за линейное время (корректность б/д).
- 57. (2) Поиск пары ближайших точек на плоскости за $\mathcal{O}(N \log N)$.
- 58. (2) Поиск пары самых удаленных точек на плоскости за $\mathcal{O}(N \log N)$.