МФТИ

Алгоритмы и структуры данных, осень 2022

Программа экзамена

Всюду, где уместно и не сказано иное, пункт программы подразумевает формулировку решаемой задачи, описание алгоритма, доказательство его корректности и анализ асимптотики.

1. Асимптотические обозначения: O, Ω, Θ. Независимость от стартового индекса.
2. Сумма на отрезке в статическом массиве: префиксные суммы.
3. Проверка вхождения числа в отсортированный массив: бинарный поиск.
4. Доказательство формулы: $\log(n!) = \Theta(n \log n)$.
5. Нижняя оценка на число сравнений в сортировке сравнениями.
6. Сортировка слиянием (Merge Sort).
7. Поиск числа инверсий в массиве.
8. Нерекурсивная реализация сортировки слиянием.
9. Быстрая сортировка (Quick Sort). Асимптотика — б/д.
10. Поиск k-й порядковой статистики с выбором случайного пивота (Quick Select). Асимптотика — б/д.
11. Детерминированный алгоритм поиска k-й порядковой статистики за $O(n)$, где n — длина массива.
12. Детерминированный алгоритм быстрой сортировки за $O(n \log n)$, где n — длина массива.
13. Стабильная сортировка подсчётом. Сортировка пар чисел.
14. Структура данных стек: реализация на указателях, использование `std::stack`.
15. Поиск ближайшего меньшего/большего слева/справа в статическом массиве.
16. Поддержка минимума в стеке.
17. Реализация очереди на двух стеках.
18. Поддержка минимума в очереди.
19. Двоичная куча: определение и представление в массиве. Требование кучи.
20. Операции `siftUp` и `siftDown` с доказательством корректности.
21. Выражение `insert`, `getMin`, `extractMin` и `decreaseKey` через `siftUp` и `siftDown`.
22. Построение кучи (`heapify`) за линейное время (сходимостью ряда можно пользоваться б/д).
23. Сортировка кучей с привлечением $O(1)$ дополнительной памяти (Heap Sort).
24. Технические сложности и их преодоление для операции `decreaseKey` в куче.
25. Удаление из кучи по значению.
26. Удаление из кучи по указателю.
27. Биномиальное дерево, биномиальная куча: определение.
28. Операции `merge`, `insert`, `getMin`, `extractMin` и `decreaseKey` в биномиальной куче.
29. Амортизационный анализ, учётное время работы: определение.
30. Метод монеток (бухгалтерский учёт).
31. Структура данных вектор, реализация на массиве и оценка асимптотики.
32. Метод потенциалов.
33. Sparse Table: модельная задача, построение за $O(n \log n)$, ответ на запрос за $O(1)$.
34. Дерево отрезков: модельная задача. Обработка запросов с доказательством времени работы.
35. Дерево отрезков: двоичный спуск, поиск k-го нуля на отрезке массива за $O(\log n)$.
36. Дерево отрезков, отложенные операции: присвоение константы на отрезке, операция `push`.
37. Количество чисел на отрезке, значения которых лежат в отрезке: Fractional Cascading.
38. Персистентный массив.
39. Персистентное дерево отрезков.
40. Количество чисел на отрезке, значения которых лежат в отрезке: решение с персистентным деревом отрезков.
41. Динамическое дерево отрезков.

42. Онлайн vs. оффлайн: сжатие координат.
43. Онлайн vs. оффлайн: дерево поиска оффлайн.
44. Онлайн vs. оффлайн: количество чисел на отрезке, значения которых лежат в отрезке.
45. Дерево Фенвика: классическая задача, операции `update` и `getSum`.
46. Обобщение дерева Фенвика на бóльшие размерности. Изменение асимптотики.
47. Дерево Фенвика: массовые операции увеличения на отрезке и запрос суммы на отрезке.
48. Дерево Фенвика: массовые операции увеличения на прямоугольнике и запрос суммы на прямоугольнике.
49. Дерево Фенвика деревьев Фенвика.
50. Хеш-таблицы. Хеш-функции. Коллизии. Универсальное и k-независимое семейства хеш-функций.
51. Хеширование цепочками. Реализация операций `find`, `insert`, `erase`.
52. Совершенное хеширование.
53. Фильтр Блума. Алгоритм и оптимальные значения параметров (б/д).
54. Хеш-таблицы с открытой адресацией. Реализация операций `find`, `insert`, `erase`. Теоремы о времени работы для линейного пробирования, двойного хеширования.
55. Дерево поиска: определения и операции (без реализации) `find`, `insert`, `erase`, а также опциональные `merge` и `split`.
56. Наивное дерево поиска, обработка операций.
57. AVL-дерево: определение.
58. Оценка глубины AVL-дерева на n вершинах.
59. Устранение дисбаланса в AVL-дереве для случая $\Delta(a) = -2$.
60. AVL-дерево: реализация операций `insert` и `erase`.
61. Splay-дерево: определение и практическая значимость.
62. Splay-дерево: операции `zig`, `zig-zig` и `zig-zag`, операция `splay`.
63. Амортизированное время работы операции `splay` с помощью метода потенциалов.
64. Splay-дерево: реализация `insert`, `erase` и `find`, связь с операцией `splay`, оценка времени работы.
65. Splay-дерево: реализация `merge` и `split`, связь с операцией `splay`, оценка времени работы.
66. B-дерево: определение и практическая значимость.
67. Оценка глубины B-дерева на n ключах при фиксированном параметре t.
68. Реализация операции `insert` в B-дереве.
69. Реализация операции `erase` в B-дереве.
70. Декартово дерево: определение и теорема о глубине (б/д).
71. Реализация операций `merge` и `split` в декартовом дереве.
72. Выражение `insert` и `erase` в декартовом дереве через `merge` и `split`.
73. Декартово дерево по неявному ключу: в массиве вставить, удалить элемент, поменять местами подмассивы.
74. Красно-чёрное дерево: определение.
75. Оценка глубины красно-чёрного дерева на n ключах.
76. [Можно пользоваться официальной шпаргалкой с разбором случаев] Реализация операции `insert` в красно-чёрном дереве.
77. [Можно пользоваться официальной шпаргалкой с разбором случаев] Реализация операции `erase` в красно-чёрном дереве.
78. Сравнительный анализ различных реализаций дерева поиска: наивное, AVL-, splay-, B-, декартово и красно-чёрное дерево.