МФТИ, ФПМИ

Алгоритмы и структуры данных, осень 2021

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

42. Персистентное дерево отрезков.
43. Количество чисел на отрезке, значения которых лежат в отрезке: решение с персистентным деревом отрезков.
44. Динамическое дерево отрезков.
45. Онлайн vs. оффлайн: сжатие координат.
46. Онлайн vs. оффлайн: дерево поиска оффлайн.
47. Онлайн vs. оффлайн: количество чисел на отрезке, значения которых лежат в отрезке.
48. Дерево Фенвика: классическая задача, операции `update` и `getSum`.
49. Обобщение дерева Фенвика на бóльшие размерности. Изменение асимптотики.
50. Обратное дерево Фенвика: максимум на отрезке и изменение (увеличение) в точке (`update` — без реализации).
51. Дерево Фенвика деревьев Фенвика.
52. Дерево поиска: определения и операции (без реализации) `find`, `insert`, `erase`, а также опциональные `merge` и `split`.
53. Наивное дерево поиска, обработка операций.
54. AVL-дерево: определение.
55. Оценка глубины AVL-дерева на n вершинах.
56. Устранение дисбаланса в AVL-дереве для случая $\Delta(a) = -2$.
57. AVL-дерево: реализация операций `insert` и `erase`.
58. Splay-дерево: определение и практическая значимость.
59. Splay-дерево: операции `zig`, `zig-zig` и `zig-zag`, операция `splay`.
60. Амортизированное время работы операции `splay` с помощью метода потенциалов.
61. Splay-дерево: реализация `insert`, `erase` и `find`, связь с операцией `splay`, оценка времени работы.
62. B-дерево: определение и практическая значимость.
63. Оценка глубины B-дерева на n ключах при фиксированном параметре t.
64. Реализация операции `insert` в B-дереве.
65. Реализация операции `erase` в B-дереве.
66. Декартово дерево: определение и теорема о глубине (б/д).
67. Реализация операций `merge` и `split` в декартовом дереве.
68. Выражение `insert` и `erase` в декартовом дереве через `merge` и `split`.
69. Декартово дерево по неявному ключу: в массиве вставить, удалить элемент, узнать сумму на отрезке.
70. Красно-чёрное дерево: определение.
71. Оценка глубины красно-чёрного дерева на n ключах.
72. [Можно пользоваться официальной шпаргалкой с разбором случаев] Реализация операции `insert` в красно-чёрном дереве.
73. [Можно пользоваться официальной шпаргалкой с разбором случаев] Реализация операции `erase` в красно-чёрном дереве.
74. Сравнительный анализ различных реализаций дерева поиска: наивное, AVL-, splay-, B-, декартово и красно-чёрное дерево.