

МФТИ, ФПМИ  
Алгоритмы и структуры данных, 2-й семестр, весна 2022  
Программа экзамена, ПМ

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

1. Асимптотические обозначения:  $O$ ,  $\Omega$ ,  $\Theta$ . Независимость от стартового индекса.
2. Структура данных стек: реализация на указателях, использование `std::stack`.
3. Поиск ближайшего меньшего/большого слева/справа в статическом массиве.
4. Поддержка минимума в стеке.
5. Реализация очереди на двух стеках.
6. Поддержка минимума в очереди.
7. Нижняя оценка на число сравнений в сортировке сравнениями.
8. Сортировка слиянием (Merge Sort).
9. Поиск числа инверсий в массиве.
10. Быстрая сортировка (Quick Sort). Асимптотика — б/д.
11. Поиск  $k$ -й порядковой статистики с выбором случайного пивота (Quick Select). Асимптотика — б/д.
12. Детерминированный алгоритм поиска  $k$ -й порядковой статистики за  $O(n)$ , где  $n$  — длина массива.
13. Детерминированный алгоритм быстрой сортировки за  $O(n \log n)$ , где  $n$  — длина массива.
14. Стабильная сортировка подсчётом. Сортировка пар чисел.
15. Цифровая сортировка (LSD).
16. Двоичная куча: определение и представление в массиве. Требование кучи.
17. Операции `siftUp` и `siftDown` с доказательством корректности.
18. Выражение `insert`, `getMin`, `extractMin` и `decreaseKey` через `siftUp` и `siftDown`.
19. Построение кучи (`heapify`) за линейное время (сходимостью ряда можно пользоваться б/д).
20. Биномиальное дерево, биномиальная куча: определение.
21. Операции `merge`, `insert`, `getMin`, `extractMin` и `decreaseKey` в биномиальной куче.
22. Амортизационный анализ, учётное время работы: определение.
23. Метод монеток (бухгалтерский учёт).
24. Структура данных вектор, реализация на массиве и оценка асимптотики методом монеток.
25. Sparse Table: модельная задача, построение за  $O(n \log n)$ , ответ на запрос за  $O(1)$ .
26. Дерево отрезков: модельная задача. Обработка запросов с доказательством времени работы.
27. Дерево отрезков: двоичный спуск, поиск  $k$ -го нуля на отрезке массива за  $O(\log n)$ .
28. Дерево отрезков, отложенные операции: присвоение константы на отрезке, операция `push`.
29. Динамическое дерево отрезков.
30. Дерево поиска: определения и операции (без реализации) `find`, `insert`, `erase`, а также опциональные `merge` и `split`.
31. Наивное дерево поиска, обработка операций.
32. AVL-дерево: определение.
33. Оценка глубины AVL-дерева на  $n$  вершинах.
34. Устранение дисбаланса в AVL-дереве для случая  $\Delta(a) = -2$ .
35. AVL-дерево: реализация операций `insert` и `erase`.
36. Декартово дерево: определение и теорема о глубине (б/д).
37. Реализация операций `merge` и `split` в декартовом дереве.
38. Выражение `insert` и `erase` в декартовом дереве через `merge` и `split`.
39. Декартово дерево по неявному ключу: в массиве вставить, удалить элемент, узнать сумму на отрезке.
40. Пять шагов для решения задачи на динамическое программирование.

41. Задача о кузнечике (набор максимальной суммы на массиве). Неоптимальность жадного алгоритма.
42. Задача о черепашке (набор максимальной суммы на таблице). Динамика “назад” и “вперёд”.
43. Задача о наибольшей общей подпоследовательности.
44. Задача о наибольшей возрастающей подпоследовательности: решение за  $O(n^2)$ .
45. Задача о наибольшей возрастающей подпоследовательности: решение за  $O(n \log n)$  с помощью дерева отрезков или дерева Фенвика со сжатием координат.
46. Задача о рюкзаке: решения с динамикой по весу или по стоимости, что лучше выбрать. Восстановление ответа.
47. Бинарное возведение чисел и матриц в степень, асимптотика.
48. Подсчёт  $n$ -го числа Фибоначчи (по модулю  $m$ ) за  $O(\log n)$ .
49. Кодирование подмножеств  $\{0, 1, \dots, n - 1\}$  с помощью масок. Процедура извлечения бита (bit).
50. Операции над множествами (масками): объединение, пересечение, разность. Реализация в программе. Проверка, что одна маска является подмножеством другой. Проверка, что число является степенью двойки.
51. Задача о самом дешёвом гамильтоновом пути: решение за  $O(2^n \cdot n^2)$ .
52. Задача о максимальной клике: решения за  $O(2^n \cdot n^2)$ ,  $O(2^n \cdot n)$  и  $O(2^n)$ .
53. Подсчёт всех значений  $b(mask) = \max_{s \subset mask} a(s)$  для данного набор значений  $a(0), \dots, a(2^n - 1)$  за  $O(2^n \cdot n)$ .
54. Задача о максимальной клике: решение за  $O(2^{n/2} \cdot n)$ . Опционально: решение за  $O(2^{n/2})$ .
55. Хеш-таблицы: постановка задачи (запросы `insert`, `erase`, `find`), хеш-функции, коллизии. Разрешение коллизий с помощью цепочек.
56. Универсальное семейство хеш-функций. Теорема о времени работы, если хеш-функция выбирается из универсального семейства. Пример универсального семейства.
57. Совершенное хеширование. Постановка и решение за  $O(n)$  предподсчёта в среднем и  $O(1)$  на запрос.
58. Определения неориентированного и ориентированного графов, пути, (вершинно) простого пути, рёберно простого пути. Связь вершинной и рёберной простоты. Определение цикла, рёберно простого цикла, (вершинно) простого цикла. Определение достижимости между вершинами, простота пути. Определение связности.
59. Отношение сильной связности между вершинами. Компоненты сильной связности. Сильно связный граф.
60. Поиск в глубину: алгоритм `dfs` на ориентированном графе. Лемма о белых путях.
61. Поиск в глубину: множество посещаемых вершин, поиск цикла, достижимого из  $s$ , проверка на ацикличность.
62. Топологическая сортировка ориентированного ациклического графа: определение и алгоритм поиска (с доказательством корректности). Число путей в ориентированном ациклическом графе.
63. Алгоритм Косарайю. Корректность и время работы.
64. Постановка и решение задачи 2SAT (применение алгоритма выделения компонент сильной связности).
65. Алгоритм `dfs` на неориентированном графе. Дерево обхода `dfs`. Классификация рёбер на древесные и обратные. Проверка связности и ацикличности. Компоненты связности.
66. Мосты, точки сочленения. Введение функции `ret`. Критерий того, что ребро является мостом.
67. Насчёт `ret` в неориентированном графе, нахождение мостов.
68. Нахождение точек сочленения в неориентированном графе.
69. Определение эйлерова цикла. Критерий наличия эйлерова цикла в неориентированном графе.
70. Реализация алгоритма поиска эйлерова цикла.
71. Определение кратчайшего расстояния в невзвешенном/взвешенном графе.
72. Поиск в ширину: алгоритм `bfs` с доказательством корректности.

73. Алгоритм 0-K-bfs.
74. Двусторонний bfs.
75. Алгоритм Дейкстры. Условия применимости, доказательство корректности. Реализации за  $O(n^2)$ ,  $O(m \log n)$ ,  $O(m + n \log n)$ .
76. Алгоритм  $A^*$ : определения функций  $f, g, h$ ; реализация.
77. Вырожденные случаи в алгоритме  $A^*$ :  $h \equiv 0$ ,  $h(v) = \text{dist}(v, t)$ .
78. Допустимые и монотонные эвристики в алгоритме  $A^*$ . Примеры монотонных эвристик на разных сетках.
79. Формулировка работоспособности (корректность и время работы) алгоритма  $A^*$  в случае монотонной, допустимой или произвольной эвристики. Доказательство для монотонного случая.
80. Алгоритм Флойда: поиск попарных кратчайших расстояний в графе без отрицательных циклов. Реализация, асимптотика.
81. Восстановление ответа (пути) в алгоритме Флойда.
82. Алгоритм Форда—Беллмана: поиск кратчайших расстояний от одной вершины до всех. Реализация, асимптотика (в случае отсутствия отрицательных циклов).
83. Алгоритм Форда—Беллмана: нахождение кратчайших расстояний от одной вершины до всех в случае наличия отрицательных циклов.
84. Остовный подграф, остовное дерево. Минимальный остов. Лемма о безопасном ребре.
85. Алгоритм Прима: доказательство корректности и реализации за  $O(n^2)$ ,  $O(m \log n)$ ,  $O(m + n \log n)$ .
86. Система непересекающихся множеств (СНМ). Виды запросов. Эвристика по рангу, эвристика сжатия путей. Асимптотика ответа на запрос при использовании обеих эвристик (б/д).
87. Асимптотика ответа на запрос в СНМ при использовании только эвристики по рангу.
88. Алгоритм Крускала: корректность, реализация, асимптотика.
89. Определение паросочетания в произвольном графе, двудольного графа, увеличивающего пути.
90. Лемма об устройстве неориентированного графа, в котором степени всех вершин не превосходят двух.
91. Теорема Бержа.
92. Алгоритм Куна. Корректность, реализация, асимптотика.
93. Лемма об отсутствии увеличивающих путей из вершины при отсутствии таких путей относительно меньшего паросочетания.
94. Определения независимого множества, вершинного покрытия. Связь определений.
95. Алгоритм поиска максимального независимого множества и минимального вершинного покрытия в двудольном графе с помощью разбиения на доли  $L^-, L^+, R^-, R^+$  (с доказательством). Теорема Кёнига.
96. Определения сети, потока, величины потока, остаточной сети. Пример, почему нельзя обойтись без обратных рёбер.
97. Определения разреза, величины разреза, величины потока через разрез. Лемма о равенстве величины потока и величины потока через разрез.
98. Лемма о связи величины произвольного потока и величины произвольного разреза.
99. Теорема Форда—Фалкерсона.
100. Алгоритм Форда—Фалкерсона. Корректность, асимптотика. Пример сверхполиномиального (от размера входа) времени работы.
101. Алгоритм Эдмондса—Карпа. Корректность.
102. Лемма о возрастании  $\text{dist}(s, v)$  между последовательными итерациями алгоритма Эдмондса—Карпа.
103. Лемма о числе насыщений ребра в алгоритме Эдмондса—Карпа. Асимптотика этого алгоритма.
104. Задача о разбиении коллектива на две группы с минимизацией суммарного недовольства.
105. Определение слоистой сети, блокирующего потока. Алгоритм Диница, доказательство корректности.
106. Реализация алгоритма Диница. Асимптотика.

107. Первая теорема Карзанова о числе итераций алгоритма Диница.
108. Быстродействие алгоритма Диница в единичных сетях.
109. Алгоритм Хопкрофта—Карпа поиска максимального паросочетания в двудольном графе. Корректность и асимптотика.
110. Min cost flow: постановка задачи. Критерий минимальности стоимости потока величины  $k$  (б/д). Алгоритм поиска потока величины  $k$  минимальной стоимости. Асимптотика.
111. Потенциалы Джонсона. Поиск min cost  $k$ -flow с помощью алгоритма Дейкстры.
112. Определение дерева, его свойства (б/д). Определение центроида в дереве. Алгоритм поиска центроида в дереве. Лемма о количестве центроидов.
113. Определение изоморфизма графов. Алгоритм проверки изоморфности двух ориентированных или неориентированных деревьев за  $O(n \log n)$ .
114. Задача LCA. Постановка, решение с помощью двоичных подъёмов.
115. Задача LCA. Решение с помощью эйлера обхода.
116. Задача LCA. Алгоритм Фарах-Колтона и Бендера.