

## Вопросы

1. **dfs.** Способы хранения графа. Оценка времени работы. Поиск путей, компонент связности и сильной связности, циклов в ор. и неор. графах. Topsort. Конденсация графа.
2. **dfs.** Поиск компонент вершинной и реберной двусвязности, мостов и точек сочленения.
3. **bfs.** Время работы. Поиск кратчайших путей в невзвешенных графах, 0-1 графах, 1-k графах, 0-1-k графах. Кратчайшие пути от  $v$  до  $A$ , от  $A$  до  $v$ , от  $A$  до  $B$ .
4. **Дейкстра.** Алгоритм, реализации за  $O(V^2)$  и за  $O(E \log V)$ . Потенциалы: замена функции весов на неотрицательную.
5. **Форд-Беллман.** Алгоритм и три его реализации: за  $O(VE)$ , с break, с циклической очередью. Поиск отрицательного цикла.
6. **Флойд.** Алгоритм, док-во корректности, реализация за  $O(V^3)$ , восстановление пути. Поиск отрицательного цикла.
7. **MST.** Алгоритм Прима и Краскала, время работы. Поиск остова с минимальным «max весом ребра» за  $O(E \log E)$ .
8. **Эйлеровость.** Эйлеров путь, цикл, признак существования, алгоритм поиска. Использование для покраски ребер двудольного регулярного графа.
9. **Гамильтоновость.** Поиск гамильтоновых пути и цикла за  $O(2^V V^2)$ . Поиск пути перебором. Эвристики min степени и связности. Жадное решение задачи про обход конем шахматной доски.
10. **Циклы.** Цикл минимальной длины во взвешенном и невзвешенном графах за  $O(E \cdot FindPath)$  и  $O(V \cdot FindPath)$ . Цикл минимального среднего веса (за  $O(VE \cdot \log)$  с док-вом и за  $O(VE)$  без док-ва).
11. **Задачи на графах.** Диаметр дерева. Дополнить граф до связного. Дополнить граф до сильносвязного. Найти путь из  $v$  в  $u$  такой, что max вес ребра минимален. Найти путь max длины в графе (свести к задаче про min путь). Проверить, является ли граф деревом.
12. **Dynamic Programming.** Время работы. Кол-во путей, min путь, max путь. Кол-во кратчайших путей. Игры на ациклических графах.
13. **Ретроанализ.** Оптимальные стратегии на графах с циклами. Алгоритм, время работы. Подсчет длины игры при оптимальной игре обоих.
14. **Функция Гранди.** Определение. Подсчет за  $O(E)$ . LOSE/WIN через функцию Гранди. Док-во факта про прямую сумму игр и хог.
15. **Задачи на Гранди.** Hacking Bush. Nim. Nim + кучку можно делить попалам. Nim + кучку можно делить на произвольное число слагаемых: (а) показать, что предыдущее решение точно не работает, (б) рассказать решение для  $\sum a_i \leq 50$ .
16. **Функция Смита.** Определение функции. Выражение LOSE/WIN/DRAW через посчитанную функцию Смита. Факт про сумму игр и хог (без док-ва). Алгоритм построения за  $O(VE)$ .
17. **Перебор.** Оптимизация полного перебор. Предподсчет. Отсечение по времени. Меморизация. Отсечение по ответу. Совмещение последних двух оптимизаций.
18. **Перебор.** Жадность, перебор с сортировкой ребер. Выбор первых  $k$  ребер. Iterative Deepening, Iterative Increasing of  $K$ .
19. **RSQ, RMQ.** Формулировки задач. Массив (матрица) не меняется. Решение задач RSQ и RMQ на подотрезке и на подпрямоугольнике. RSQ решить частичными суммами, RMQ — с помощью Sparse Table.
20. **LCA. Online.** Двоичные подъемы, решение за  $[N \log N, \log N]$ . Сведение к RMQ, решение за  $[N \log N, 1]$ . Как для двух вершин в дереве за  $O(1)$  в Online проверить, что одна — предок другой?
21. **Функции на путях дерева.** Деление путей на вертикальные. Сумма = LCA +  $O(1)$ , минимум = двоичные подъемы.
22. **Дерево отрезков.** Хранение в массиве. Решение RSQ и RMQ. «+=» и «=» на отрезке.  $K$ -е число в множестве.
23. **Корневая оптимизация.** Решение задач RMQ и RSQ для меняющегося массива разбиением массива на части. Решение задачи RSQ для меняющегося массива отложенными операциями.

24. **Фенвик + бонус.** Дерево Фенвика, 2D дерево Фенвика. Минимум на отрезке фиксированной длины в online за  $O(1)$ .
25. **Сведение LCA-RMQ.** LCA  $\rightarrow$  RMQ за  $O(n)$ . RMQ  $\rightarrow$  LCA за  $O(n)$ . Построение декартового дерева за  $O(n)$ .
26. **Декартово дерево.** Определение, единственность, оценка глубины. Split и Merge. Выражение через них Insert, Delete. Randomized BST (у можно не хранить).
27. **Декартово дерево.** Функция на отрезке. К-й элемент в дереве. Неявный ключ. Операции с деревом по неявному ключу: вставка в середину, удаление из середины, поменять местами два куска, reverse на отрезке.
28. **СНМ.** Решение: get за  $O(1)$  и join за  $O(N \log N)$  в сумме. Обобщение идеи сливания множества «меньшее к большему» до произвольных структур данных. Пример полученного обобщения: сливаемыми кучами и сливаемые деревьями поиска.
29. **СНМ.** Решение со сжатием путей: get и join работают за  $O(\alpha)$  (обратная функция Аккермана). Док-во того, что  $M$  запросов работают в сумме за  $O((M + N) \log^* N)$ .
30. **Хэш-таблицы.** Списки (Add, Del, Find). Прямая адресация (Add, Find). Что происходит с обеими структурами при переполнении? Как бороться с переполнением? Оценка вероятности того, что хэш-таблица на списках работает «долго»
31. **BST.** Несбалансированные и сбалансированные деревья поиска. Вращения: малое и большое. Постоянная перебалансировка, возможные инварианты (высоты, размеры). Перебалансировка, когда  $size = 2^k$ . Splay-дерево. Показать, что BST умеет все тоже, что и дерево отрезков.
32. **Persistent.** Persistent tree, persistent array, persistent world. Garbage collection.
33. **2D-запрос.** Кол-во точек на прямоугольнике. Решение в Offline сканирующей прямой. Решение в Online с помощью 2D-дерева (дерево отрезков сортированных массивов). Решение в Online задачи «кол-во элементов массива с  $i$  по  $j$ , имеющих значение с  $x$  по  $y$ ». Эквивалентность двух последних задач. Случай, когда массив меняется (дерево отрезков декартовых деревьев).
34. **Heavy-Light decomposition.** Общее решений задач на путях дерева. Покрытие дерева вертикальными путями. Доказать время ответа на запрос  $O(\log^2 N)$ .
35. **Задачи.** Подотрезок max суммы за  $O(N)$ . Подматрица max суммы за  $O(N^2)$ . Persistent СНМ. Структура, умеющая делать Add, Del, Find за  $O(1)$  и ListAll за  $O(\text{кол-во элементов в структуре})$ .
36. **Бор.** Бор, сжатый бор, суф. дерево. Построение бора и сжатого бора. Построение суф. дерева за  $O(N^2)$  времени с использованием  $O(N)$  памяти. Сортировка слов бором за  $O(\sum |Word|)$ .
37. **Бор.** Способы хранения ребер бора (массив, список, хэш-таблица). Решение задачи о поиске словарных слов в тексте за время  $O(|Text| \cdot \max |Word|)$ . Решение задачи о количестве различных подстрок суф. деревом для  $N \leq 20\,000$ .
38. **Хэши.** Полиномиальный хэш. Отсутствие коллизий. Сравнение подстрок на равенство за  $O(1)$ . Сравнение подстрок на больше-меньше за  $O(\log N)$ . Поиск подстроки в тексте за  $O(|T| + |S|)$  с использованием  $O(1)$  дополнительной памяти.
39. **Хэши.** Решение задач (поиск подстроки в тексте за  $O(N)$ , общая подстрока за  $O(N \log N)$ , наибольший подпалиндром за  $O(N \log N)$  и за  $O(N)$ , количество различных подстрок строки за  $O(N^2)$  с  $O(N)$  памяти).
40. **Z, Prefix.** Z и Prefix функции, определение и построение за линейное время. Решения задач поиска строки в тексте, поиска периода строки, min циклического сдвига строки.
41. **Суф.дерево.** Укконен за  $O(N)$  (без док-ва времени работы).
42. **Суф.дерево.** Решение задач (кол-во подстрок, max по длине строка, встречающаяся как подстрока два раза, наибольшая общая подстрока, max подпалиндром, наибольший общий подпалиндром, K-я лексикографически подстрока, поиск подстроки в тексте).
43. **Суф.массив.** Построение за  $O(N \log^2 N)$ , построение LCP за  $O(N \log N)$ .
44. **Суф.массив.** Построение за  $O(N \log N)$ . LCP за  $O(n)$ .
45. **Суф.массив.** Решение задач (кол-во подстрок, max по длине строка, встречающаяся как подстрока два раза, наибольшая общая подстрока, поиск подстроки в тексте).
46. **Суф.структуры.** Преобразование суф. массива в суф. дерево и обратно за  $O(N)$ .

47. **Ахо-Корасик.** Построение автомата за  $O(\sum |Word|)$ . Использование автомата за  $O(|Text|)$ . Как для каждого слова в словаре за  $O(|Text| + \sum |Word|)$  подсчитать количество вхождений слова в текст?
48. **ЛСР.** Определение. Подсчет динамикой за  $O(N^2)$  всех ЛСР. Подсчет ЛСР для любой пары хэшами за  $O(\log N)$ . ЛСР любой пары за  $O(RMQ)$  с помощью суф. массива. ЛСР любой пары за  $O(LCA)$  с помощью суф. дерева.
49. **Потоки.** Определения: поток, разрез, остаточная сеть. Отношение размеров потока и разреза. Теорема Форда-Фалкерсона.
50. **Потоки.** Алгоритмы построения: Форд-Фалкерсон, Эдмондс-Карп, Масштабирование потока. Времена работы. Вершинный поток. Нахождение паросочетания с помощью потока.
51. **Потоки.** Алгоритм Диница за  $O(V^2E)$ . Описание алгоритма, время работы. Скрещивание с масштабированием.
52. **Потоки.** Решение задач (восстановление матрицы по суммам в строках и столбцах, восстановление турнирной таблице, если какие-то матчи уже проставлены и для каждой команды известно кол-во очков в конце). [L..R] поток (избытки и недостатки).
53. **MinCost потоки.** Поиск MinCost потока размера 0. Поиск MinCost потока размера  $k + 1$ . Док-во корректности. Использование Дейкстры вместо Форд-Беллмана (потенциалы).
54. **Паросочетания.** Определения: паросочетание, совершенное паросочетание, независимое множество, контролирующее множество.  $\forall M, C : |M| \leq |C|$ . Лемма о дополняющем пути в произвольном графе.
55. **Паросочетания.** Алгоритм Куна построения паросочетания. Алгоритм построения по готовому шах паросочетанию min контролирующего и шах независимого множеств за  $O(E)$ .
56. **Паросочетания.** Решение задач. (Max clique,  $A : |A| - |Im(A)| = max$ , покрытие ациклического графа путями, задача о такси, замощение грида минимальным кол-вом палок  $1 \times N$ ).
57. **Раскраски.** Общие факты про вершинную и реберную покраску. Лемма Холла. Существование совершенного паросочетания в двудольном регулярном графе.
58. **Раскраски.** Покраска ребер  $k$ -регулярного двудольного графа за время  $O(k \cdot Matching)$ , за время  $O(\log k \cdot Matching)$ . Покраска ребер произвольного двудольного графа.
59. **Паросочетания.** Неточный рандомизированный алгоритм построения паросочетания в произвольном графе. Тест, на котором алгоритм не работает с вероятностью  $\frac{1}{2}$ . Построение паросочетания в произвольном графе динамикой за  $O(2^N \cdot N)$
60. **Гамильтоновость.** Теоремы Дирака и Ore. Алгоритм построения гамильтонова цикла в насыщенном графе за  $O(N^2)$ . Док-во теорем и корректности алгоритма.
61. **Турниры.** Поиск гамильтоновых цикла и пути в турнире за  $O(N^2)$ . Критерии существования. Построение цикла фиксированной длины  $k$  для  $\forall k \geq 3$  (циклы можно сперва построить за  $O(N^3)$ , а после оптимизировать до  $O(N^2)$ ).
62. **Планарность.** Теорема Эйлера с док-вом. Лемма  $E \leq 3V - 6$  с док-вом, ее случай для двудольного графа. Не планарность  $K_5$  и  $K_{3,3}$ . Формулировка теоремы Куратовского.
63. **Планарность.** Алгоритм Демукрона укладки графа с док-вом корректности и полиномиальности времени работы. Факты про существование упадок в  $R_3, S_2$ , прямыми отрезками, на гриде.
64. **2-SAT.** Формулировки задач SAT, 3-SAT, 2-SAT. Жадное решение 2-SAT за  $O(VE)$ . Решение 2-SAT за  $O(E)$ .
65. **Матроиды.** Три аксиомы матроидов. Примеры матроидов с док-вом того, что указанные множества образуют матроид: (а) лес, (б) набор линейно независимые вектора, (в) множество вершин первой доли двудольного графа, которые можно покрыть паросочетанием. Алгоритм построения Базы Матроида. Алгоритм Радо-Эдмондса построения Базы min веса.