Содержание

1 Неделя 1

1. Найти наибольшую возрастающую подпоследовательность в массиве за $O(n \log n)$, используя дерево отрезков.
2. Вычислить количество инверсий в перестановке за $O(n \log n)$, используя дерево отрезков.

3-7. Для каждой задачи из этого блока сформулируйте, для какого множества значений и какой ассоциативной операции нужно построить дерево отрезков. Структуру самого дерева и алгоритм выполнения операций изменять нельзя.

Есть массив a из n целых чисел и две операции: присвоить значение: $a_i = x$, и одна из следующих:

3. Найти минимум на отрезке от l до r, а так же число элементов, равных этому минимуму.
4. Найти минимум на отрезке от l до r, а так же индекс самого левого элемента, равного этому минимуму.
5. Найти значение суммы $a_l - a_{l+1} + a_{l+2} - a_{l+3} + \ldots \pm a_{r-1}$.
6. Найти значение суммы $a_l + 2a_{l+1} + 3a_{l+2} + \ldots + (r-l)a_{r-1}$.
7. Найти на заданном отрезке $[l, r)$ подотрезок $[l_1, r_1)$ $(l \leqslant l_1 \leqslant r_1 \leqslant r)$, сумма на котором максимальна (достаточно вывести эту сумму, но можно и отрезок тоже).

8-10. В задачах из этого блока нужно добавить новую операцию в дерево отрезков.

Есть массив a из n целых чисел и две операции: присвоить значение: $a_i = x$, и одна из следующих:

8. Найти минимальное i для которого $a_i \geqslant k$.
9. Вывести все i для которых $a_i \geqslant k$ за время $O(x \log n)$, где x — размер ответа.

10. Найти минимальное i на отрезке от l до r, для которого $a_i \geqslant k$.

11-13. Есть парковка на n мест. Каждое место может быть занятым или свободным. Нужно обрабатывать операции: пометить место как занятое/свободное, и одна из следующих:

11. Найти число свободных мест на отрезке от l до r.
12. Найти k-е по порядку свободное место.
13. Найти свободное место, ближайшее к i.

14-15. Есть строка из n круглых скобок. Нужно обрабатывать запросы: 1) изменить i-ю скобку, 2):

14. Проверить, является ли правильной скобочной последовательностью подстрока с l до r.
15. Найти наибольший префикс подстроки с l до r, который является правильной последовательностью.

16. Есть перестановка. Запрос: даны l, r, x, y, нужно вычислить количество элементов на отрезке $[l, r]$ со значениями от x до y. Дано q запросов, нужно на все вывести ответы. Время $O((n + q) \log n)$.
17. То же самое, но на запросы нужно отвечать в онлайне. То есть, следующий запрос приходит после ответа на предыдущий. Время на предподсчет $O(n \log n)$, время на запрос $O(\log n)$.
18. Есть массив, элементы равны 1 или 2. Отвечать на запросы:
    - Изменить значение элемента.
    - Дано l и r, найти длину НВП $a[l..r]$.
19. То же самое, до элементы могут равняться 1, 2 и 3.

2 Неделя 2

Есть массив a из n значений типа boolean. Нужно обрабатывать запросы за $O(\log n)$.

20. - Присвоить значение x всем элементам отрезка;
    - Найти ближайшую к i единицу.
21. - Изменить значение всех элементов отрезка на противоположное;
    - Найти число единиц на отрезке.
22. - Присвоить значение x всем элементам отрезка;
    - Выполнить `for (i = l .. r - 1) : a[i] = a[i] and a[i - 1]`;
    - Выполнить `for (i = l .. r - 1) : a[i] = a[i] or a[i - 1]`;
    - Найти число единиц на отрезке.
23. - Присвоить значение x всем элементам отрезка;
    - Найти число непрерывных отрезков из единиц.
24. - Присвоить значение x всем элементам отрезка;
    - Найти самый длинный непрерывный отрезок из единиц.

Есть массив a из n целых чисел. Нужно обрабатывать запросы за $O(\log n)$.

25.
- Присвоить значение x всем элементам отрезка;
- Изменить элементы отрезка $a_i = -a_i$;
- Найти сумму на отрезке.

26.
- Присвоить значение x всем элементам отрезка;
- Изменить элементы отрезка $a_i = -a_i$;
- Найти максимум на отрезке.

27.
- Присвоить значение x всем элементам отрезка;
- Изменить элементы отрезка $a_i = -a_i$;
- Найти подотрезок с максимальной суммой.

28.
- Присвоить значение x всем элементам отрезка;
- Прибавить x ко всем элементам отрезка;
- Найти значение элемента.

29.
- Присвоить значение x всем элементам отрезка;
- Прибавить x ко всем элементам отрезка;
- Найти сумму на отрезке.

30.
- Изменить элементы отрезка $a_i = \max(a_i, x)$;
- Найти минимум на отрезке.

31.
- Изменить элементы отрезка $a_i = \max(a_i, x)$;
- Найти максимум на отрезке.

32.
- Изменить элементы отрезка $a_i = \max(a_i, x)$;
- Изменить элементы отрезка $a_i = \min(a_i, x)$;
- Найти значение a_i.

33.
- Изменить элементы отрезка $a_i = a_i + x$;
- Изменить элементы отрезка $a_i = \min(a_i, x)$;
- Найти значение a_i.

34.
- Присвоить значение x всем элементам отрезка;
- Найти самый длинный отрезок из одинаковых чисел.

35.
- Изменить элементы отрезка $a_i = a_i + x \cdot i + y$;
- Найти сумму на отрезке.

36.
- Присвоить элементам отрезка значения $a_i = x \cdot i + y$;
- Найти максимум на отрезке.

37.
- Присвоить элементам отрезка значения $a_i = x \cdot i + y$;
- Найти НОД чисел на отрезке.

38.
- Присвоить элементам отрезка значения $a_i = x \cdot i + y$;
- Найти отрезок с максимальной суммой.

39.
- Присвоить $a_i = x$;
- Вычислить $\sum\limits_{i=l}^{r} a_{i \oplus x}$ (даны l, r, x).

3 Неделя 3

40. Постройте дерево Фенвика по заданному массиву за $O(n)$.
41. Постройте дерево Фенвика по заданному массиву без дополнительной памяти за $O(n)$.
42. Дано дерево Фенвика, восстановите исходный массив без дополнительной памяти за $O(n)$.
43. Как изменятся время работы и объем используемой памяти разреженной таблицы, если хранить отрезки длиной не 2^k, а x^k $(x > 2)$?
44. Добавить в дерево Фенвика операцию, которая находит максимальный префикс массива, на котором сумма не больше x (все значения неотрицательные) за $O(\log n)$.
45. Есть шахматное поле $n \times n$. Обрабатывать запросы: 1) добавить/удалить ладью, 2) найти число клеток в заданном прямоугольнике, которые не бьются ни одной ладьей. Оба запроса за $O(\log n)$.
46. Последовательность f_i вычисляется по следующим правилам: $f_{-1} = f_0 = 1$, $f_i = (a_i \cdot f_{i-1} + b_i \cdot f_{i-2}) \pmod M$. Нужно обрабатывать запросы: 1) для заданного i изменить числа a_i и b_i на x и y (и пересчитать последовательность), 2) найти значение f_i. Оба запроса за $O(\log n)$.
47. Есть полоса из n клеток, в некоторых могут находиться препятствия. Есть кузнечик, который может прыгать на одну, две или три клетки. Обрабатывать запросы: 1) добавить/удалить препятсвие, 2) найти число способов добраться из клетки x в клетку y. Оба запроса за $O(\log n)$.
48. Есть два массива a и b. Нужно обрабатывать запросы: 1) скопировать участок массива a в масиив b (то есть, сделать $b_{y+q} = a_{x+q}$ для всех q от 0 до $k-1$) 2) найти значение b_i. Оба запроса за $O(\log n)$.
49. Есть лес из n деревьев, стоящих в ряд, высота i-го дерева равна a_i. На лес последовательно падает m метеоритов. Метеорит с силой p_j, упавший рядом с деревом x_j, ломает все еще не сломанные деревья i, для которых $a_i \leqslant p_j - |x_j - i|$. Найдите для каждого дерева, какой по счету метеорит его сломает за $O((n+m)\log n)$.
50. Про некоторый массив a из n чисел известно m свойств вида $\max(a_l..a_r) = x$. Постройте массив, удовлетворяющий всем свойствам или скажите, что это невозможно.
51. Про некоторый массив a из n чисел известно m свойств вида $\sum(a_l..a_r) = x$. Постройте массив, удовлетворяющий всем свойствам или скажите, что это невозможно.
52. В этой задаче можно использовать только обычное дерево Фенвика. Есть массив a из n чисел. Нужно обрабатывать запросы: 1) прибавить x ко всем элементам отрезка, 2) найти значение a_i.
53. В этой задаче можно использовать несколько деревьев Фенвика. Есть массив a из n чисел. Нужно обрабатывать запросы: 1) прибавить x ко всем элементам отрезка, 2) найти сумму a_i на отрезке.
54. Есть массив a (он не меняется). Отвечать на запросы: найти сумму элементов на отрезке $a[l..r]$, значения которые лежат в интервале от x до y.
55. Есть массив a (он не меняется). Отвечать на запросы: есть ли элемент, который встречается на отрезке $a[l..r]$ больше раз, чем все остальные в сумме (то есть занимает больше половины отрезка)?

56. Байтландия представляет собой $n+1$ городов, расположенных вдоль прямой дороги. Города занумерованы числами от 0 до n по возрастанию координат. Столица имеет номер 0 и расположена в точке с координатой 0. Город с номером i имеет координату x_i. Телефоны еще не изобрели, поэтому на случай чрезвычайной ситуации в каждом городе находится специальный гонец. Если случается чрезвычайная ситуация, гонец в городе i собирается a_i секунд, после чего бежит по направлению к столице, тратя b_i секунд на один километр пути. Когда гонец добегает до другого города, он может либо пробежать мимо, либо зайти в город и передать послание местному гонцу (в этом случае второй гонец также собирается, и затем бежит). Найдите для каждого города, за какое минимальное время послание из него может быть доставлено в столицу.

4 Неделя 4

Двумерные структуры

57. Есть несколько непересекающихся прямоугольников. Можно за один ход перепрыгнуть с одного прямоугольника на другой, если есть вертикальный или горизонтальный отрезок, соединяющий эти прямоугольники и не пересекающий другие прямоугольники. Найдите для каждого прямоугольника список прямоугольников, на которые с него можно прыгнуть $O(n \log n + ans)$.

58. Есть несколько прямоугольников. Они расположены так, что их границы не пересекаются, но один прямоугольник может быть внутри другого. Постройте дерево вложенности прямоугольников за $O(n \log n)$.

59. Таблица $n \times n$, обрабатывать запросы: 1) прибавить значение к прямоугольнику, 2) найти значение в ячейке. Оба запроса за $O(\log^2 n)$.

60. Таблица $n \times n$, обрабатывать запросы: 1) прибавить значение к прямоугольнику, 2) найти сумму значений в прямоугольнике. Оба запроса за $O(\log^2 n)$. (Решите с помошью нескольких двумерных деревьев Фенвика.)

61. Таблица $n \times n$. После предподсчета за $O(n^2)$ отвечать на запросы: по данным (x, y, d) находить сумму в прямоугольном треугольнике с вершинами в (x, y), $(x+d, y)$ и $(x, y+d)$.

62. Та же задача, но есть еще операция изменить элемент таблицы.

63. Представьте, что вам нужно построить дерево отрезков размера M и совершить с ним n операций. При этом n сильно меньше M. Попробуем сэкономить память, храня только интересные узлы. Покажите, как получить структуру, в которой $O(n)$ узлов.

64. Примените идеи из предыдущей задачи к двумерному дереву. Какие результаты получатся?

65. Есть n точек в трехмерном пространстве, координаты целые и не больше k. Найти последовательность точек максимальной длины такую, что точки в ней монотонно возрастают по всем трем координатам. За $O(k^2 + n \log n \log^2 k)$.

66. Есть n точек в трехмерном пространстве, координаты целые и не больше k. Найти последовательность точек максимальной длины такую, что точки в ней монотонно возрастают по всем трем координатам. За $O(n(\log n + \log^2 k))$.

67. На очередных тренировочных сборах команд программистов состоялось три соревнования. Теперь каждая команда считает себя сильнее всех команд, которых она обыграла хотя бы на одном из этих соревнований. Сколько существует пар команд, в которых каждая команда считает себя сильнее другой? $O(n \log^3 n)$.

68. Манхеттенское расстояние на плоскости задается как сумма расстояний по координатам $|x_1 - x_2| + |y_1 - y_2|$. Дана таблица $n \times n$, в некоторых клетках которой находятся фишки.

Нужно по запросу (x, y, r) за $O(\log^2 n)$ находить число фишек, находящихся на манхеттенском расстоянии не больше r от точки (x, y).

69. Есть массив из чисел a_i. Поступают запросы:

    - Даны l, r, k, b. Применить $a_i = \min(a_i, k \cdot i + b)$ ко всем элементам на отрезке $[l, r]$.
    - Даны l, r. Вычислить минимум на отрезке $[l, r]$.

    Время $O(\log^2 n)$ на запрос.

5 Неделя 5

Деревья поиска

70. Напишите рекурсивную процедуру, выводящую элементы дерева поиска в отсортированном порядке, за время $O(n)$.
71. Напишите нерекурсивную процедуру, выводящую элементы дерева поиска в отсортированном порядке, за время $O(n)$ с $O(1)$ дополнительной памяти (у узлов есть указатели на родителей).
72. Докажите, что нет алгоритма, который строит дерево поиска по заданному массиву из n элементов быстрее, чем за $O(n \log n)$ в худшем случае.
73. Для каждого узла x посчитайте число $w(x)$, равное числу узлов в его поддереве (включая сам x). Время $O(n)$.
74. Используя вычисленные значения $w(x)$, научитесь находить k-й по возрастанию элемент дерева. Время $O(H)$ (H — высота дерева).
75. Используя $w(x)$, научитесь находить по заданному ключу x число элементов, меньших x. Время $O(H)$ (H — высота дерева).
76. По данному узлу дерева найдите следующие k узлов в отсортированном порядке за время $O(H + k)$.
77. Приведите пример дерева, в котором средняя глубина узла (среднее расстояние от узла до корня) $O(\log n)$, а высота дерева (максимальное расстояние от узла до корня) — $\omega(\log n)$ (асимптотически больше).
78. Приведите пример АВЛ-дерева, в котором при добавлении нужно будет совершить процедуру балансировки более чем в одном узле.
79. Приведите пример АВЛ-дерева, в котором при удалении нужно будет совершить процедуру балансировки более чем в одном узле.
80. Приведите пример двух АВЛ-деревьев, хранящих одно и то же множество элементов, но имеющих разную высоту.
81. Двоичное дерево поиска называется красно-черным, если каждая его вершина раскрашена либо в красный, либо в черный цвет, родитель любой красной вершины черный, и путь до любого листа содержит одно и то же количество черных вершин. Докажите, что высота такого дерева $O(\log n)$.
82. Двоичное дерево поиска называется сбалансированным по весу, если для любой вершины v выполняется $w(v) \geqslant \lfloor \alpha \cdot w(parent(v)) \rfloor$, где $w(v)$ — число вершин в поддереве, α — какая-то положительная константа. Докажите, что высота такого дерева $O(\log n)$.
83. Проверить, что заданное дерево является корректным деревом поиска. Время $O(n)$.

84. Пусть в дереве для каждого узла высота его детей отличается не более чем на 5. Правда ли, что высота такого дерева $O(\log n)$?
85. Пусть в дереве для каждого узла высота его детей отличается не более чем в два раза. Правда ли, что высота такого дерева $O(\log n)$?
86. Приведите пример двух 2-3-деревьев, хранящих одно и то же множество элементов, но имеющих разную высоту.
87. По аналогии с 2-3 деревом можно сделать X-Y дерево, в котором у каждого узла, кроме листьев (и, возможно, корня), от количество детей лежит в отрезке X до Y, включительно. Какие условия нужно наложить на X и Y, чтобы можно было эффекивно работать с таким деревом?

6 Неделя 6

Деревья поиска 2

88. Дан массив пар x, y, отсортированный по x. Постройте по нему декартово дерево за $O(n)$.
89. Дан массив пар x, y. Все x различны, а y могут совпадать. Проверить, что декартово дерево можно построить единственным образом.
90. Дан массив пар x, y. Все x различны, а y могут совпадать. Найти число способов построить декартово дерево.
91. Покажите, как на базе дерева поиска по неявному ключу сделать СНМ со временем работы $O(\log n)$.
92. Реализуйте в декартовом дереве по неявному ключу операцию $splitAfter(x)$, которая разделяет дерево, содержащее узел x, на деревья до x, включительно, и после x. Считайте, что размеры поддеревьев **не посчитаны**, зато у всех элементов есть ссылка на родителя в дереве.
93. Покажите, что если взять обычное дерево поиска без всяких балансировок и добавлять в него элементы в порядке убывания y, то получится декартово дерево.
94. Пусть у дерева поиска нет узлов с одним ребенком (то есть у каждой внутренней вершины ровно два ребенка). Правда ли, что высота такого дерева $O(\log n)$?
95. Пусть в дереве размер (число вершин) для любого поддерева равен $2^k - 1$ для какого-то целого k. Правда ли, что высота такого дерева $O(\log n)$?
96. Дан массив чисел от 1 до n. Научитесь с помощью декартова дерева по неявному ключу обрабатывать запросы за $O(\log n)$: 1) развернуть отрезок массива от l до r задом наперед, 2) найти i-й элемент.
97. Дан массив чисел от 1 до n. Научитесь с помощью декартова дерева по неявному ключу обрабатывать запросы за $O(\log n)$: 1) для данных l и r поменять местами пары элементов: l и $l+1$, $l+2$ и $l+3$, ..., $r-1$ и r, 2) найти i-й элемент.
98. Будем строить декартово дерево, не храня ключи y. В тех местах кода, где производится сравнение y, выберем случайный вариант с вероятностью 50%. Покажите, что после такого изменения некоторая последовательность действий может привести к тому, что высота дерева (вернее, ее матожидание) будет $\Omega(n)$.
99. В дереве поиска, в отличие от дерева отрезков, исходные элементы множества хранятся не только в листьях, но и в промежуточных узлах. Иногда это неудобно, поэтому делают другую версию дерева поиска: исходные элементы множества хранятся только в листьях, а в промежуточных вершинах хранится максимальный ключ в поддереве. Покажите, как осуществлять операции поиска и добавления элемента в таком дереве.

100. Покажите, что структура дерева из предыдущего задания изоморфна обычному дереву поиска на $n-1$ элементах, а значит его можно балансировать всеми теми же способами.
101. Есть n окружностей. Они не пересекаются, но могут лежать внутри друг друга. Постройте дерево вложенности за $O(n \log n)$.
102. Есть несколько кругов. Они не пересекаются, но могут касаться. Постройте для каждого круга список тех, с кем он касается, за $O(n \log n)$.
103. Есть несколько многоугольников, в них суммарно n вершин. Они не пересекаются, но могут лежать внутри друг друга. Постройте дерево вложенности за $O(n \log n)$.

7 Неделя 7

Splay дерево

104. Пусть вы сделали splay для вершины, которая находилась на глубине H. Посчитайте, как изменится сумма глубин вершин на этом пути после операции.
105. Покажите, что если делать splay просто каждый раз делая zig, то есть последовательность из n операций, которая работат дольше $n \log n$.
106. Покажите, что если делать splay, каждый раз беря два ближайших предка и делая поворот верхнего, а затем нижнего ребра (как в zig-zig), то есть последовательность из n операций, которая работат дольше $n \log n$.
107. Пусть в splay дереве хранятся числа от 1 до n. Будем делать m операций: `splay(1)`, `splay(n)`, `splay(1)`, `splay(n)`, ... За какое суммарное время они будут работать?
108. Пусть в splay дереве размера n вы совершаете много действий над небольшим подмножеством из k элементов. Как будет выглядеть дерево? За какое время будут работать операции?
109. Пусть в AVL дереве хранятся числа от 1 до n. Покажите, что если делать операцию find по порядку для всех чисел от 1 до n, то суммарное время работы будет больше $O(n)$.
110. Пусть в splay дереве хранятся числа от 1 до n. Покажите, что если делать операцию find по порядку для всех чисел от 1 до n, то суммарное время работы будет $O(n)$.
111. Пусть в splay дереве хранятся числа от 1 до n. К дереву делают m запросов, суммарное число запросов к элементу i равно p_i. Покажите, что суммарное время запросов $O(m + \sum p_i \log \frac{m}{p_i})$ (подсказка: на лекции в доказательстве у всех вершин были веса 1 и $s(v)$ равнялось количеству вершин в поддереве, теперь нужно выбрать для вершин разные веса $w(v)$, и $s(v)$ обозначить сумму весов вершин в поддереве, чтобы $r(x)$ был таким, каким нужно).
112. Пусть в splay дереве хранятся числа от 1 до n. К дереву делают m запросов x_i. Покажите, что для любого элемента f суммарное время запросов $O(m + n \log n + \sum \log(|x_i - f| + 1))$.
113. Пусть в splay дереве хранятся числа от 1 до n. К дереву делают m запросов x_i. При этом известно, что $|x_i - x_{i+1}| \leqslant D$. Покажите, что суммарное время запросов $O(m \log D)$.
114. Покажите, как делать операцию `split` в splay дереве. За какое время она будет работать? (не забудьте посчитать изменение потенциала).
115. Покажите, как делать операцию `merge` в splay дереве. За какое время она будет работать? (не забудьте посчитать изменение потенциала).
116. Покажите, как построить splay дерево по заданному отсортированному массиву за $O(n)$. (не забудьте посчитать изменение потенциала).

8 Неделя 8

Scapegoat Tree и другие задачи

117-118. Countdown Tree это дерево поиска, похожее на Scapegoat Tree. Оно устроено следующим образом. Когда создается узел v, в него записывается число, равное $\alpha \cdot w(v)$. Каждый раз, когда в поддереве узла совершается операция, это число уменьшается на 1. Когда число в узле становится равным 0, поддерево узла перестраивается с нуля.

117. Покажите, что высота такого дерева в каждый момент времени $O(\log n)$.
118. Покажите, амортизированное время операции добавления на таком дереве $O(\log n)$.

119-120. Dynamite Tree это еще одно дерево поиска, похожее на Scapegoat Tree. Оно устроено следующим образом. В каждом узле храним значение $w(v)$. Когда совершается операция в поддереве вершины v, мы бросаем монетку, и с вероятностью $1/w(v)$ ломаем это поддерево и перестраиваем его с нуля.

119. Покажите, что матожидание высоты такого дерева в каждый момент времени $O(\log n)$.
120. Покажите, матожидание времени операции добавления на таком дереве $O(\log n)$.

121. Как сделать операцию `merge` в Scapegoat дереве за амортизированное $O(\log n)$?
122. Как сделать операцию `split` в Scapegoat дереве за амортизированное $O(\log n)$?
123. Было дерево поиска, его узлы выписали в порядке «корень, левое поддерево, правое поддерево». Восстановите дерево.
124. Дано n элементов, для каждого из них известна частота обращений f_i. Постройте оптимальное дерево поиска, то есть дерево, минимизирующее величину $\sum f_i \cdot d_i$, где d_i — глубина узла i.
125. Дано двоичное дерево из n узлов. Убрать из него минимальное число узлов, чтобы оно стало корректным Scapegoat деревом для данного α.
126. Покажите, как на базе дерева поиска сделать мультимножество, то есть хранить множество, в котором может быть несколько объектов с одинаковым ключом.
127. Есть два двоичных дерева. За одно действие можно поменять местами детей у любого узла. Проверить, что одно дерево можно получить из другого.
128. Двоичное дерево. Расставить в вершины дерева различные числа длиной $O(\log n)$ бит так, чтобы по числам в двух данных вершинах можно было (с помощью хеш-таблиц и битовой магии) за $O(1)$ понять, какое число находится в их ближайшем общем предке.
129. Дано двоичное дерево поиска. За какое минимальное количество малых поворотов можно сделать это дерево корректно сбалансированным AVL деревом? $o(n^2)$.

9 Неделя 9

LCA и двоичные подъемы

130. Дано дерево, отвечать на запросы «найти число ребер на пути от v до u» за $O(\log n)$ (ну или сразу за $O(1)$).
131. Дано дерево, на каждом ребре написано число. Отвечать на запросы «найти сумму на пути от v до u» за $O(\log n)$ (ну или сразу за $O(1)$).
132. Дано дерево, на каждом ребре написано число. Отвечать на запросы «найти минимум на пути от v до u» (веса не меняются) за $O(\log n)$.

133. Научитесть вычислять любую ассоциативную функцию на пути в дереве за $O(\log n)$.
134. Дано дерево, у каждого ребра есть длина. Научитесь отвечать на запросы: «найти ближайшего предка v, расстояние до которого не меньше l» за $O(\log n)$.
135. Дано дерево, у каждого ребра есть длина. Научитесь отвечать на запросы: «найти середину пути от v до u» за $O(\log n)$ (середина пути — это либо вершина, либо точка на ребре, во втором случае можете просто найти нужное ребро).
136. Запросы: «дано множество из k вершин, найдите их ближайшего общего предка» за $O(k \log n)$.
137. Запросы: «дано множество из k вершин, найдите число вершин, которые являются предком хотя бы для одной из них» $O(k \log n)$.
138. Пусть дерево иногда меняется. Добавим две операции: «создать вершину v и подвесить ее к u» и «удалить лист v». Покажите, что можно пересчитывать двоичные подъемы без потери производительности структуры.
139. Запросы: «найти расстояние от v до u», «поменять вес ребра», оба за $O(\log n)$.
140. Запросы: «найти длину пересечения двух путей» за $O(\log n)$.
141. В дереве покрашены вершины. Для каждой вершины найти количество различных цветов в ее поддереве $O(n \log n)$.
142. Запросы: 1) посчитать LCA двух вершин и 2) переподвесить поддерево вершины u куда-то к другой вершине (все за $O(\log n)$).
143. Начиная в корне дерева, посетить k заданных вершин и вернуться в корень. найти минимальную стоимость такого пути за $O(k \log n)$.
144. Сделать структуру данных, которая хранит массив чисел, и позволяет делать два вида операций: 1) добавить новый элемент в конец массива (`push_back`), найти минимум на отрезке. Обе за $O(1)$.

10 Неделя 10

Heavy-light декомпозиция и другое

145. Какое максимальное и минимальное число путей может быть в Heavy-light декомпозиции дерева?
146. Будем выбирать в качестве тяжелого ребра ребро, ведущее в поддерево максимальной высоты (а не максимального веса). Какое максимальное число легких ребер может при этом быть на пути до корня?
147. Научитесь обрабатывать такие запросы за полилог: 1) изменить вес ребра, 2) найти самый далекий от корня лист.
148. Есть дерево, каждое ребро может быть либо включено, либо выключено. Научитесь обрабатывать такие запросы за полилог: 1) изменить состояние всех ребер на пути от u до v, 2) найти число компонент связности по включенным ребрам.
149. Есть дерево, каждое ребро может быть либо включено, либо выключено. Научитесь обрабатывать такие запросы за полилог: 1) изменить состояние ребра, 2) найти самый длинный путь от корня по включенным ребрам.
150. Есть дерево, каждое ребро может покрашено в один из 7 цветов. Научитесь обрабатывать такие запросы за полилог: 1) покрасить все ребра на пути из u до v в цвет c, 2) найти число ребер каждого цвета.

151. Есть дерево, вершины бывают включены и выключены. Запросы: 1) выключить вершину (обратно не включаются), 2) найти для данной вершины ближайшего включенного предка. Быстрее, чем за $O(\log n)$.

152. Есть дерево, в вершинах написаны числа. Запросы: 1) изменить значение в вершине; 2) вычислить сумму на пути; 3) вычислить сумму в поддереве.

153. Есть сеть железных дорог в виде дерева из двухколейных путей. Для каждого пути известно время, за которое его проходит поезд. Нужно отвечать на запросы: «Первый поезд отправляется из A в B во время X, второй — из C в D во время Y, правда ли, что есть момент времени, в который поезда едут по одному и тому же ребру?».

Link-Cut Tree и другое

154. Научитесь за $O(\log n)$ искать LCA в link-cut дереве.

155. Добавить в link-cut дерево операцию `connected(u, v)`, проверяющую, что вершины u и v находятся в одном дереве.

156. Посмотрите, как меняется потенциал после операций `link` и `cut`, убедитесь, что амортизированное время работы этих операций $O(\log n)$.

157. Link-cut дерево, ребра бывают двух цветов. Проверять, что в пути до корня ребра чередуются.

158. Link-cut дерево, вершины бывают n цветов. Проверять, что в пути до корня нет двух соседних вершин одного цвета.

159. Link-cut дерево, на каждом ребре написан его вес. Находить на пути от v до корня ближайшее к v ребро с весом не больше d.

160. Есть граф из n вершин, изначально пустой. В него добавляются ребра. После каждого добавления нужно пересчитать вес минимального остовного дерева.

161. Попробуем поддерживать heavy-light декомпозицию дерева после операций `link` и `cut`. У скольких ребер может измениться состояние после одной операции?

162-163. Можно ли быстро найти все ребра, у которых изменилось состояние? (Чтобы было проще, будем считать, что тяжелое ребро — это то, для которого $size(u) \geqslant size(v)/2$).

    162. Найти ребра, которые были легкими, а стали тяжелыми.

    163. Найти ребра, которые были тяжелыми, а стали легкими.

164. Дано дерево, в каждой вершине записано число. Запросы: 1) дана вершина v и число x. Прибавить ко всем потомкам u вершины v число $x - d(u, v)$, где $d(u, v)$ — расстояние между вершинами, 2) найти значение числа в вершине. Отвечать на оба запроса за $O(\log n)$.

165. Дана таблица $d[i, j]$. Построить такое взвешенное дерево, чтобы расстояние от вершины i до вершины j было равно $d[i, j]$.

166. Есть дерево. За одну операцию можно отрезать от него любой лист. Посчитать число различных множеств вершин, которые можно отрезать за k таких операций на заданном дереве.

167. Есть дерево. За одну операцию можно отрезать от него любой лист. Посчитать, сколько есть способов совершить k таких операций на заданном дереве.

168. Есть дерево из n вершин. По этому дереву ходят n фуникулеров, каждый фуникулер двигается от вершины x_i до вершины y_i и обратно, с остановками в каждой вершине, при этом вершина x_i — предок вершины y_i. Вам нужно за $O(\log n)$ отвечать на запросы: найти минимальное число фуникулеров, чтобы добраться из вершины v в вершину u.

169. Есть дерево из n вершин. По этому дереву ходят n автобусов, каждый автобус двигается от вершины x_i до вершины y_i и обратно, с остановками в каждой вершине. Вам нужно за $O(\log n)$ отвечать на запросы: есть ли автобус, на котором можно доехать от вершины v до вершины u.

11 Неделя 11

Центроидная декомпозиция

170. (Пофикшенная 152) Есть дерево, в вершинах написаны числа. Запросы: 1) прибавить число на пути; 2) прибавить число в поддереве; 3) вычислить сумму на пути; 4) вычислить сумму в поддереве.
171. Пусть в дереве степени всех вершин не больше 3. Докажите, что есть ребро, при удалении которого дерево распадется на компоненты, размер каждой из которых не больше $\frac{2}{3}n + 1$.
172. Докажите, что в дереве не более двух центроидов.
173. Задано подвешенное дерево. Найдите для каждого поддерева его центроид. Время $O(n)$.
174. Дано дерево, каждое ребро имеет длину. Научитесь за полилог находить число вершин на расстоянии не больше D от заданной.
175. Дано дерево, каждое ребро имеет длину. Научитесь за полилог выполнять операции: 1) поменять длину ребра, 2) найти число вершин на расстоянии не больше D от заданной.
176. Дано дерево, каждая вершина имеет положительный вес. Научитесь за полилог выполнять операции: 1) поменять вес вершины, 2) найти путь максимального веса, проходящий через заданную вершину.
177. Дана дорожная сеть в виде дерева, которую вечно ремонтируют. Ремонт производится этапами. Этап (v, d) состоит в том, что ремонтируются все ребра на расстоянии не больше d от вершины v. Отвечать на запросы: провести этап ремонта, узнать, когда последний раз ремонтировалось ребро.
178. Дано дерево, над ним проводят секретные эксперименты. Каждый эксперимент (v, d) состоит в том, что в вершине v распыляют вонючее вещество с вонючестью d. При этом для всех $x < d$ уровень вони во всех вершинах на расстоянии x увеличивается на $d - x$. Отвечать на запросы: провести эксперимент, найти текущий уровень вони в вершине.
179. Дано дерево, вершины покрашены в черный и белый цвета. Отвечать на запросы: изменить цвет вершины; найти суммарное расстояние от v до вершин того же цвета.
180. Дано дерево, вершины покрашены в черный и белый цвета. Отвечать на запросы: изменить цвет вершины; найти ближайщую от v вершину того же цвета.
181. Дано дерево путей между городами, для каждого ребра известна стоимость проезда, также известна стоимость колбасы в каждой вершине. Научитесь за полилог искать, как дешевле всего купить колбасу из заданной вершины, с учетом стоимости проезда.
182. Дано дерево путей между городами, для каждого ребра известно время проезда. Происходят n событий, про каждое событие известно, в какой вершине и в какое время оно происходит. Нужно узнать, какое максимальное число событий можно посетить. Время $O(n \log^2 n)$. (подсказка: сначала придумайте решение за $O(n^2)$ с помощью динамического программирования)
183. Дано дерево, в вершинах написаны числа $c[v]$. Обрабатывать запросы: изменить значение $c[v]$, посчитать сумму $\sum_u \sum_v c[lca(u, v)]$. Время $O(\log n)$.
184. Дан массив из чисел. Отвечать на запросы: дан отрезок $[l, r]$, найдите максимальное k такое, что на данном отрезке можно выделить подпоследовательность $1, 2, \ldots, k$.

185. Дано дерево, ребра имеют стоимости. Для каждого x от 0 до $n-1$ посчитать минимальную суммарную стоимость ребер, которые нужно удалить, чтобы степени всех вершин были не больше x. Время $O(n \log n)$. (подсказка: сначала придумайте решение для фиксированного x с помощью динамического программирования)

186. Рассмотрим клетчатую плоскость, по ней можно перемещаться в соседнюю клетку по стороне. Дана связная фигура из черных клеток без дыр (то есть, множество черных клеток связно, и множество белых клеток связно). Отвечать на запросы: найти длину кратчайшего черного пути между двумя черными клетками. Время $O(\log n)$. (подсказка: посчитать отдельно число ходов по вертикали и по горизонтали, для этого разбить фигуру на части таким образом, чтобы они образовали дерево).

187. Дано дерево, каждое ребро имеет длину. В каждой вершине находится бомба, про каждую бомбу известен радиус взрыва. Если бомбу взорвать, она сдетонирует бомбы, которые находятся от нее на расстоянии не больше радиуса взрыва, те тоже взорвутся и сдетонируют какие-то еще, и т. д. Узнайте, какое минимальное число бомб нужно взорвать вручную, чтобы в результате взорвались все бомбы. Время $O(n \log^2 n)$. (подсказка: сначала придумайте решение за $O(n^2)$, построив граф, кто кого взрывает)

12 Неделя 12

Геометрия

188. Дано множество выпуклых многоугольников на плоскости. Никакие два многоугольника не пересекаются по границе. Постройте дерево вложенности за $O(s \log s)$, где s — суммарное количество вершин во всех многоугольниках.

189. Дан простой многоугольник (не обязательно выпуклый), вам даны его вершины в порядке обхода против часовой стрелки. Постройте выпуклую оболочку множества вершин многоугольника за $O(n)$.

190. Дана простая ломаная (последовательность точек, соединенных отрезками, не имеющая самопересечений или самокасаний). Постройте выпуклую оболочку множества вершин ломаной за $O(n)$.

191. Дано множество точек. Найдите две наиболее удаленные точки за $O(n \log n)$.

192. Дано множество точек. Найдите равносторонний треугольник минимального размера, покрывающий все точки за $O(n \log n)$.

193. Дан простой многоугольник. Проверить, что отрезок, соединяющий две вершины является диагональю. $O(n)$ на один отрезок.

194. Дан простой многоугольник. Посчитайте количество триангуляций. $O(n^3)$.

195. Дан выпуклый многоугольник. Поступают запросы — точки. Нужно для каждой точки за $O(\log n)$ отвечать, лежит ли она внутри мноугольника.

196. На плоскости дано n кругов одинакового радиуса. Найти точку, которая принадлежит им всем, или сказать, что такой нет. Время $O(n)$.