АиСД у2022. Третий семестр

Домашние задания М3134-М3137

⟨Версия от 25 декабря 2023 г.⟩

Темы

Неделя 1. Обход в глубину

Устная часть

1.1. Раскраской графа в k цветов называется назначение каждой вершине одного из k цветов. Раскраска называется правильной, если нет пары одноцветных вершин, соединенных ребром. Дан неориентированный граф. Покрасьте его правильным образом в два цвета, либо определите, что это невозможно. Время работы — $\mathcal{O}(n+m)$.

1.2. Есть n человек. Некоторые пары людей дружат. Требуется разбить всех людей на две команды из $\frac{n}{2}$ человек таким образом, чтобы каждая пара друзей попали в разные команды. Время работы — $\mathcal{O}(n^2)$.

1.3. Рассмотрим следующий алгоритм:

```
visited = 0

dfs(v):
    visited += 1
    for (v, u) in E:
        if (visited < n):
            dfs(u)

for v = 1 ... n:
    if (visited < n):
        dfs(v)
```

Докажите или опровергните, что данный фрагмент кода, запущенный на ациклическом ориентированном графе, работает за время $\mathcal{O}(n+m)$.

1.4. Докажите или опровергните, что фрагмент кода из предыдущей задачи, запущенный на произвольном ориентированном графе, работает за время $\mathcal{O}(n+m)$.

1.5. Дан ациклический ориентированный граф. Найдите в нем простой путь, состоящий из максимально возможного количества ребер. Простым называется путь, который проходит по каждой вершине и по каждому ребру не более одного раза. Время работы: $\mathcal{O}(n+m)$.

1.6. Дан ациклический ориентированный граф. Найдите в нем длину кратчайшего пути из s в t, а также количество кратчайших путей из s в t. Время работы: $\mathcal{O}(n+m)$.

1.7. Придумайте алгоритм, который строит лексикографически минимальную топологическую сортировку за $\mathcal{O}((n+m)\log n)$.

1.8. Есть n гирек, веса которых попарно различны. Вам дана последовательность из m взвешиваний на чашечных весах: в каждом взвешивании участвовали ровно две гирьки, по одной на каждой чаше весов. Результат каждого взвешивания дает информацию, которая из двух гирек легче. Возможно, результаты некоторых взвешиваний некорректны. Определите, после какого минимального

количества взвешиваний полученная информация гарантированна стала противоречивой. Время работы: $\mathcal{O}((n+m)\log n)$.

1.9. У вас есть n дел, некоторые дела зависят от других дел: их нельзя сделать раньше тех, от которых они зависят. Всего есть m зависимостей. Граф зависимостей ацикличен. Выполнение каждого дела занимает один час. Дело под номером t является очень важным. Определите порядок выполнения дел, чтобы очень важное дело t было выполнено как можно раньше. Время работы: $\mathcal{O}(n+m)$.

1.10. Дан неориентированный граф с n вершинами и m ребрами. Требуется разбить множество вершин V на три множества A, B и C: $V = A \sqcup B \sqcup C$, таким образом, чтобы были выполнены следующие условия:

(a) Множество вершин A является путем (можно выбрать $|A|-1$ ребер, соединяющих некоторые пары вершин A, чтобы получился путь).

(b) В графе не существует ребра (u, v), такого что $u \in B$ и $v \in C$.

(c) $|B| = |C|$.

Разрешается делать некоторые из множеств A, B и C пустыми. Время работы: $\mathcal{O}(n+m)$.

Интересный факт: требуемое разбиение всегда существует.

1.11. Треугольником в неориентированном графе называется тройка вершин (u, v, w), такая что в графе существуют ребра (u, v), (u, w) и (v, w). Найдите количество треугольников в графе за $\mathcal{O}(n^3)$.

1.12. Решите предыдущую задачу за $\mathcal{O}(nm)$.

1.13. Петя решил реализовать алгоритм построения топологической сортировки следующим образом:

```
dfs(v):
    ans.add(v)
    mark[v] = true

    for (v, u) in E:
        if !mark[u]:
            dfs(u)

for v = 1 ... n:
    if !mark[v]:
        dfs(v)
```

Докажите или опровергните, что данный алгоритм на произвольном ациклическом ориентированном графе построит топологическую сортировку.

1.14. Дан ориентированный ациклический граф. Проверьте, что данный граф имеет ровно одну топологическую сортировку. Время работы: $\mathcal{O}(n+m)$.

1.15. Дан неориентированный граф. Найдите минимальный по размеру набор ребер, который нужно удалить из графа, чтобы граф стал ацикличным. Время работы: $\mathcal{O}(n+m)$.

1.16. Есть n человек, у каждого есть не более трех врагов. Отношение вражды симметрично. Разбейте людей на две группы таким образом, чтобы у каждого человека в его группе было не более одного врага. Время работы: $\mathcal{O}(n)$.

Неделя 2. КСС, конденсация графа, 2-SAT

Устная часть

2.17. Какое минимальное и максимальное количество ребер может быть в графе на n вершинах с k компонентами сильной связности? Считайте, что петли и кратные ребра запрещены.

2.18. Дан ориентированный граф. Найдите минимальное по размеру множество вершин, таких что любая вершина графа была достижима хотя бы из одной вершины из множества. Время работы: $\mathcal{O}(n+m)$.

2.19. Дан ориентированный граф. Определите, какое минимальное число ребер нужно добавить в него, чтобы он стал сильно связным. Время работы: $\mathcal{O}(n+m)$.

2.20. Дан ориентированный граф. Постройте граф, с таким же множеством вершин и минимальным числом ребер, чтобы его конденсация совпадала с конденсацией данного графа. Время работы: $\mathcal{O}(n+m)$.

2.21. Дан ориентированный граф. Для каждой вершины необходимо найти вершину с минимальным номером, достижимую из нее. Время работы: $\mathcal{O}(n+m)$.

2.22. Дан ориентированный граф. По некоторым ребрам можно перемещаться, только имея пропуск, а по остальным ребрам можно перемещаться, даже не имея пропуска. Изначально вы находитесь в вершине s, и у вас есть пропуск. Выясните, существует ли возможность потерять пропуск в графе: это значит, что вы должны пройти некоторый путь из вершины s, затем оставить пропуск в какой-то вершине, после этого пройти еще некоторый путь из этой вершины, после чего у вас не должно быть возможности вернуться в вершину, в которой вы оставили пропуск. Время работы: $\mathcal{O}(n+m)$.

2.23. Дано число n и m пар чисел (a_i, b_i), где $1 \le a_i, b_i \le n$. Требуется построить граф на n вершинах с минимальным количеством ребер, чтобы для любой пары существовал путь из вершины a_i до вершины b_i. Время работы: $\mathcal{O}(n+m)$.

2.24. *Турниром* называется ориентированный граф, в котором каждая пара вершин соединена ребром (в одну или другую сторону). *Гамильтонов путь* — это путь в графе, проходящий по каждой вершине ровно один раз. Докажите, что в любом турнире существует гамильтонов путь.

2.25. *Гамильтонов цикл* — это цикл в графе, проходящий по каждой вершине ровно один раз. Докажите, что в любом сильно связном турнире существует гамильтонов цикл.

2.26. Докажите, что если в каждой вершине турнира число входящих и исходящих ребер одинаково, то турнир сильно связен.

2.27. Дана задача 2-SAT, причем гарантируется, что для каждой скобки вида $x \vee y$ существует парная скобка $\bar{x} \vee \bar{y}$, а также для каждой скобки вида $\bar{x} \vee y$ существует парная скобка $x \vee \bar{y}$. Найдите решение такой задачи 2-SAT, в котором будет минимальное количество переменных, равных единице. Время работы: полином от n.

2.28. Дана задача 2-SAT, требуется найти ее лексикографически минимальное решение. Решением считается последовательность значений переменных $x_1, x_2, \ldots, x_n$, минимизировать нужно ее. Время работы: полином от n.

2.29. Дана задача 2-SAT и ее решение. Найдите следующее решение в лексикографическом порядке. Время работы: полином от n.

2.30. Есть n различных целых чисел $a_1, a_2, \ldots, a_n$, а также два целых числа x и y. Требуется разделить все числа a_i на два множества A и B таким образом, чтобы были выполнены следующие условия:

- ▷ Если $t \in A$, то $(t - x) \in A$
- ▷ Если $t \in B$, то $(t - y) \in B$

Время работы: полином от n.

Неделя 3. Мосты и точки сочленения. Двусвязность

Устная часть

3.31. Дан неориентированный граф. Необходимо определить, сколько ребер нужно добавить в него, чтобы он стал реберно двусвязным? Время работы: $\mathcal{O}(n+m)$.

3.32. Дан неориентированный связный граф. Необходимо ориентировать его ребра так, чтобы получился сильно связный граф. Время работы: $\mathcal{O}(n+m)$.

3.33. Какое максимальное число мостов может быть в графе с n вершинами?

3.34. Какое максимальное число точек сочленения может быть в графе с n вершинами?

3.35. Дан неориентированный граф. Необходимо за $\mathcal{O}(\log n)$ отвечать на запросы: «для заданных вершин u и v сколько существует ребер, по которым мы обязательно пройдем, если будем идти из u в v». Препроцессинг: $\mathcal{O}(n \log n + m)$.

3.36. Дан неориентированный граф. Необходимо за $\mathcal{O}(\log n)$ отвечать на запросы: «для заданных вершин u и v сколько существует вершин, по которым мы обязательно пройдем, если будем идти из u в v». Препроцессинг: $\mathcal{O}(n \log n + m)$.

3.37. Дан неориентированный граф. На каждом ребре записано число. Необходимо за $\mathcal{O}(\log n)$ отвечать на запросы: «для заданных вершин u и v какое максимальное число можно встретить на реберно простом пути из u в v». Препроцессинг: $\mathcal{O}(n \log n + m)$.

3.38. Дан неориентированный граф. На каждом ребре записано число. Необходимо за $\mathcal{O}(\log n)$ отвечать на запросы: «для заданных вершин u и v какое максимальное число можно встретить на вершинно простом пути из u в v». Препроцессинг: $\mathcal{O}(n \log n + m)$.

3.39. Есть n компьютеров, некоторые m пар компьютеров соединены проводами. Также имеется очень важный файл, который можно сохранить на некоторых компьютерах, причем должно быть возможно доставить этот файл на любой компьютер, используя провода. Найдите минимальное количество компьютеров, на которые следует сохранить файл, при условии, что любой **один** провод может сломаться и перестать передавать данные, чтобы было достигнуто описанное выше условие. Время работы: $\mathcal{O}(n+m)$.

3.40. Докажите или опровергните следующие утверждения:

- ▷ Все ребра, исходящие из точки сочленения, являются мостами
- ▷ Из каждой точки сочленения исходит хотя бы один мост
- ▷ Если все ребра, исходящие из вершины, являются мостами, то эта вершина является точкой сочленения
- ▷ Если из вершины исходит хотя бы один мост, то она является точкой сочленения
- ▷ Если из вершины исходят хотя бы два моста, то она является точкой сочленения

▷ Если вершина лежит на простом цикле, то она не может быть точкой сочленения

3.41. Дан связный неориентированный граф. Найдите количество способов удалить из графа два ребра таким образом, чтобы он стал несвязным. Время работы: $\mathcal{O}(n^2)$.

3.42. Решите предыдущую задачу за $\mathcal{O}(m)$.

3.43. Назовем граф реберным кактусом, если каждое его ребро лежит не более, чем на одном простом цикле. По данному графу за $\mathcal{O}(n+m)$ определите, является ли он реберным кактусом. Если да, то найдите все его циклы.

3.44. Дан реберный кактус. Научитесь отвечать на запросы: «найти количество простых путей из вершины u в вершину v» за $\mathcal{O}(\log n)$. Препроцессинг: $\mathcal{O}(m + n \log n)$.

Неделя 4. Эйлеровость и задачи на графы

Устная часть

4.45. Дан связный неориентированный граф. Определите, какое минимальное количество ребер нужно добавить, чтобы он стал эйлеровым. Граф называется эйлеровым, если в нем существует Эйлеров цикл.

4.46. Дан (не обязательно связный) неориентированный граф. Определите, какое минимальное количество ребер нужно добавить, чтобы он стал эйлеровым.

4.47. Дан ориентированный граф. Определите, какое минимальное количество ребер нужно добавить, чтобы он стал эйлеровым.

4.48. Петя складывает цепочки из слов, где каждое следующее слово начинается на ту же букву, на которую закончилось предыдущее. Задан словарь. Определите, можно ли собрать цепочку, используя все слова из него по одному разу.

4.49. Требуется построить минимальную по длине двоичную строку s, которая сожержит в себе как подстроки (подотрезки) все возможные двоичные строки длины k. Время работы: $\mathcal{O}(2^k \cdot k)$.

4.50. Дан ориентированный граф. Найдите в нем реберно простой цикл нечетной длины.

4.51. Есть n комнат и m дверей между комнатами. Необходимо каждую дверь покрасить с одной стороны в зеленый цвет, с другой в оранжевый цвет так, чтобы для каждой комнаты количества зеленых и оранжевых дверей в комнате отличались не более чем на один. Время работы: $\mathcal{O}(n + m)$.

4.52. Дан полный неориентированный граф на n вершинах. Из него удалили m ребер. Найдите в итоговом графе компоненты связности за $\mathcal{O}(n + m)$.

4.53. Найдите правильную раскраску неориентированного графа в три цвета, если для каждой вершины дан один цвет, в который ее запрещено красить. Время работы: $\mathcal{O}(n + m)$.

4.54. Есть n лампочек, некоторые из которых включены, а некоторые выключены, и m переключателей. Каждый переключатель меняет состояние какого-то подмножества лампочек, при этом для каждой лампочки есть не более двух переключателей, которые меняют ее состояние. Проверить, можно ли выключить все лампочки. Время $\mathcal{O}(m + n)$.

4.55. Дана таблица a размера $n \times m$, в каждой клетке которой записано целое число. Мы хотим построить таблицу b размера $n \times m$, такую что выполнены следующие условия:

- ▷ Для любой строки i и столбцов j и k верно, что если $a_{i,j} < a_{i,k}$, то $b_{i,j} < b_{i,k}$, а также если $a_{i,j} = a_{i,k}$, то $b_{i,j} = b_{i,k}$
- ▷ Для любого столбца j и строк i и k верно, что если $a_{i,j} < a_{k,j}$, то $b_{i,j} < b_{k,j}$, а также если $a_{i,j} = a_{k,j}$, то $b_{i,j} = b_{k,j}$
- ▷ Максимальное число в таблице b должно быть как можно меньше

Постройте таблицу b за время $\mathcal{O}(nm)$.

Неделя 5. Обход в ширину. Алгоритмы Дейкстры и Форда-Беллмана

Устная часть

5.56. Дан ориентированный граф. Найдите все ребра, для которых существует кратчайший путь из s в t, который проходит через данное ребро.

5.57. Пусть вес пути — это не сумма весов ребер, а произведение. Модифицируйте алгоритм Дейкстры. Для каких весов он будет работать?

5.58. Пусть вес пути — это не сумма весов ребер, а максимум. Модифицируйте алгоритм Дейкстры.

5.59. Пусть вес пути — это минимум из весов ребер. Как найти кратчайший путь?

5.60. Найти путь с минимальной суммарной длиной, а среди таких — из минимального числа ребер.

5.61. Приведите пример графа с отрицательными ребрами, на котором алгоритм Дейкстры (реализация с `set`-ом) работает экспоненциально долго.

5.62. Модифицируйте BFS, чтобы искать кратчайший путь в графе, в котором каждое ребро имеет вес 0 или 1. Время работы: $\mathcal{O}(n + m)$.

5.63. Модифицируйте BFS, чтобы искать кратчайший путь в графе, в котором каждое ребро имеет целочисленный вес от 1 до k. Время работы: $\mathcal{O}(n + k \cdot m)$.

5.64. Калькулятор умеет делать две операции: $a = (a \cdot 3) \pmod M$ и $a = (a \cdot 4) \pmod M$. За сколько операций можно получить из числа a число b? Время работы: $\mathcal{O}(M)$.

5.65. Приведите пример графа, на котором алгоритм Дейкстры делает $\Omega(n^2)$ успешных релаксаций. Релаксацией называется обновление расстояния до некоторой вершины.

5.66. Есть сеть железных дорог. Про каждую дорогу известно, сколько можно получить денег, если сдать ее на металлолом. Нужно получить как можно больше денег, но чтобы из города s в город t существовал путь.

5.67. Дан граф, в котором, возможно, присутствуют отрицательные циклы. Модифицируйте алгоритм Форда-Беллмана, чтобы найти все вершины, до которых существует бесконечно малый путь из вершины s. Время работы: $\mathcal{O}(nm)$.

5.68. Дан взвешенный неориентированный связанный граф. Известно, что $m - n \le 20$. Вам нужно отвечать на запросы: найти кратчайшее расстояние между двумя заданными вершинами. Препроцессинг за $\mathcal{O}(m \log n)$, ответ на запрос за $\mathcal{O}(1)$.

5.69. Дано взвешенное дерево. Найдите расстояние между каждой парой вершин. Время работы: $\mathcal{O}(n^2)$.

5.70. Можно ли построить такой ориентированный граф, в котором одно и то же ребро лежит как на кратчайшем пути из v в u, так и на кратчайшем пути из u в v?

5.71. Дана матрица кратчайших расстояний между всеми парами вершин. Построить граф, которому она соответствует или сказать, что такого не существует.

Неделя 6. Алгоритм Флойда. Минимальное остовное дерево

Устная часть

6.72. Пупа и Лупа все перепутали, и реализовали алгоритм Флойда следующим образом:

```
for (i = 1 ... n):
    for (j = 1 ... n):
        for (k = 1 ... n):
            d[i][j] = min(d[i][j], d[i][k] + d[k][j])
```

Придумайте граф, на котором данный алгоритм неправильно построит матрицу длин кратчайших путей.

6.73. Аналогично предыдущему заданию, придумайте граф для алгоритма, в котором вершины перебираются в порядке: i, k, j.

6.74. Докажите, что если запустить алгоритм, перебирающий вершины в порядке: i, j, k, три раза, он гарантированно построит корректную матрицу расстояний.

6.75. Докажите, что если запустить алгоритм, перебирающий вершины в порядке: i, k, j, два раза, он гарантированно построит корректную матрицу расстояний.

6.76. Рассмотрим путь из u в v, на котором вес максимального ребра минимален. Докажите, что, пользуясь только ребрами из минимального остовного дерева, можно построить путь с таким же значением веса максимального ребра.

6.77. Найдите в графе второе по сумме весов минимальное остовное дерево за $\mathcal{O}(nm \log m)$.

6.78. Есть n городов. Между городами u_i и v_i можно построить дорогу, потратив на это $A \cdot l_i$ миллиардов рублей. Также в любом городе можно построить аэропорт, потратив на это B миллиардов рублей. Вам известны константы A и B, найдите минимальную сумму денег, необходимую для того, чтобы построить дороги и аэропорты так, чтобы из любого города можно было добраться в любой другой (считается, что между любой парой городов, где есть аэропорт, можно пролететь на самолете).

6.79. Найдите в графе максимальное остовное дерево.

6.80. Проверьте, что в графе существует единственное минимальное остовное дерево.

6.81. Найдите в графе остовное дерево, в котором вес максимального ребра минимален.

6.82. Для каждого ребра проверить:

- ▷ Правда ли, что есть минимальное остовное дерево, которое его содержит
- ▷ Правда ли, что есть минимальное остовное дерево, которое его не содержит

6.83. Изобретаем алгоритм Борувки!

Рассмотрим такой алгоритм:

- ▷ Для каждой вершины графа найдем ребро, исходящее из него, имеющее минимальный вес
- ▷ Добавим все найденные ребра в ответ, после чего сожмем каждую компоненту связности в одну вершину (по аналогии с тем, как мы это делали, когда сжимали компоненты реберной двусвязности)
- ▷ Будем повторять данные шаги до тех пор, пока в графе не останется одна вершина

Докажите, что данный алгоритм строит корректное минимальное остовное дерево.

6.84. Придумайте, как реализовать алгоритм Борувки за $\mathcal{O}(m \log n)$.

Неделя 7. Строки. Префикс-функция и N-функция

Устная часть

7.85. Придумайте алгоритм построения N-функции по имеющейся префикс-функции. Время работы: $\mathcal{O}(n)$.

7.86. Придумайте алгоритм построения префикс-функции по имеющейся N-функции. Время работы: $\mathcal{O}(n)$.

7.1.1 Префикс-функция

Задания из этого блока разрешается решать только при помощи префикс-функции.

7.87. *Периодом* строки s называется такая строка t, что $s = t + t + \ldots + t$. Дана строка s длины n, трубуется найти все ее периоды за $\mathcal{O}(n)$.

7.88. Строка называется *палиндромом*, если она одинаково читается как слева-направо, так и справа-налево. Дана строка s длины n, требуется найти все ее префиксы, являющиеся палиндромами за $\mathcal{O}(n)$.

7.89. Дана строка s длины n. Для каждого префикса строки нужно посчитать, сколько раз он встречается в строке как подстрока. Время работы: $\mathcal{O}(n)$.

7.90. По данной префикс-функции постройте какую-нибудь строку с такой префикс-функцией, либо определите, что такой строки не существует. Время работы: $\mathcal{O}(n)$.

7.91. Две последовательности чисел назовем *эквивалентными*, если одну можно получить из другой прибавлением одного числа ко всем элементам (например, последовательности $(3, 6, 4)$ и $(7, 10, 8)$). Дан массив a длины n и массив b длины m. Найдите в массиве a все подмассивы, эквивалентные b. Время работы: $\mathcal{O}(n + m)$.

7.92. Две строки назовем эквивалентными, если одну можно получить из другой заменой букв, при которой одинаковые буквы переходят в одинаковые, а разные — в разные (например, строки `ABACABA` и `TQTETQT` являются эквивалентными). Дана строка a длины n и строка b длины m. Найдите в строке a все подстроки, эквивалентные b. Время работы: $\mathcal{O}(n + m)$.

7.93. Дана строка s длины n. Постройте по данной строке автомат, в котором будет $n+1$ состояние (пронумеруем их числами от 0 до n). Автомат должен допускать переход из состояния i в состояние j по символу c, если длина максимального суффикса строки $s[0...i-1] + c$, совпадающего с префиксом s, равна j. Время работы: $\mathcal{O}(n\Sigma)$.

7.94. Найдите все вхождения строки s длины n в строки $t_1, t_2, \ldots, t_m$, суммарная длина которых равна m. Время работы: $\mathcal{O}(n\Sigma + m)$.

7.1.2 N-функция

Задания из этого блока разрешается решать только при помощи N-функции.

7.96. Дана строка s длины n. Найдите самую длинную подстроку строки, которая является префиксом, суффиксом и подстрокой (не префиксом и не суффиксом) s. Время работы: $\mathcal{O}(n)$.

7.97. По данной N-функции постройте какую-нибудь строку с такой N-функцией, либо определите, что такой строки не существует. Время работы: $\mathcal{O}(n)$.

7.98. Дана строка s длины n и строка t длины m. Найдите в строке t все подстроки длины n, отличающиеся от s в одном символе. Время работы: $\mathcal{O}(n+m)$.

7.99. Дана строка s длины n. Найдите количество суффиксов s, которые лексикографически меньше, чем вся строка s. Время работы: $\mathcal{O}(n)$.

7.100. Чтобы сжать длинную строку, используется следующий формат: строка `D(X)` означает, что строку X надо повторить D раз. Такие конструкции могут вкладываться друг в друга, например «`3(2(A)2(B))C`» разжимается в «`AABBAABBAABBC`». Для данной строки длины s длины n найдите самое короткое представление в таком виде. Время работы: $\mathcal{O}(n^3)$.

Неделя 8. Бор. Ахо-Корасик

Устная часть

Во всех заданиях, где это не оговорено отдельно, считайте, что алфавит имеет константный размер.

8.101. Дан набор строк s_i. Найдите строку минимальной длины, которая не является префиксом никакой из них. Время работы: $\mathcal{O}(\sum|s_i|)$.

8.102. Дан набор строк s_i. Научитесь отвечать на запросы «найти длину наибольшего общего префикса строк s_i и s_j» за $\mathcal{O}(\log(\sum|s_i|))$. Препроцессинг: $\mathcal{O}((\sum|s_i|)\log(\sum|s_i|))$.

8.103. Сожмем вершины бора, у которых только один ребенок, разрешив писать на ребрах не только символы, а целые строки. Сколько вершин и ребер будет в таком сжатом боре?

8.104. Придумайте, как хранить сжатый бор, потратив на это $\mathcal{O}(\sum|s_i|)$ дополнительной памяти.

8.105. Дана строка s. Найдите количество ее различных подстрок. Время работы: $\mathcal{O}(|s|^2)$.

8.106. Придумайте структуру данных, поддерживающую следующие операции над множеством целых чисел S:

- ▷ Добавить число x в S,
- ▷ Удалить число x из S,
- ▷ Найти такое число $y \in S$, что $x \oplus y$ максимально (здесь $\oplus$ — это побитовое исключающее ИЛИ).

Время работы на каждую операцию: $\mathcal{O}(\log C)$, где C — максимальное число x среди всех запросов.

8.107. Придумайте структуру данных, поддерживающую следующие операции над множеством строк S:

- ▷ Добавить строку x в S,
- ▷ Удалить строку x из S,
- ▷ Найти лексикографически минимальную строку $y \in S$, такую что $x \le y$.

Время работы на каждую операцию: $\mathcal{O}(|x|)$.

8.108. Постройте автомат, который принимает только строки, содержащие хотя бы одну строку s_i, как подстроку. Время работы: $\mathcal{O}(\sum|s_i|)$.

8.109. Постройте автомат, который принимает только строки, содержащие каждую строку s_i, как подстроку. Время работы: $\mathcal{O}(\sum|s_i|)$.

8.110. Дан набор строк s_i. Проверьте, существует ли строка длины n, не содержащая ни одну из строк s_i, как подстроку. Время работы: $\mathcal{O}(n \cdot \sum|s_i|)$.

8.111. Дан набор строк s_i. Проверьте, существует ли строка бесконечной длины, не содержащая ни одну из строк s_i, как подстроку. Время работы: $\mathcal{O}(\sum|s_i|)$.

8.112. Дан набор строк s_i и строка t. Нужно вырезать из строки t максимальное количество попарно непересекающихся подстрок таким образом, чтобы каждая подстрока находилась в множестве s_i (разрешается вырезать равные подстроки). Время работы: $\mathcal{O}(\sum|s_i| + |t|)$.

8.113. Даны строки s_1, s_2 и t. Найдите пару непересекающихся вхождения строк s_1 и s_2 в строке t. Время работы: $\mathcal{O}(\sum|s_i|)$.

8.114. Решите предыдущую задачу для трех строк s_1, s_2 и s_3.

8.115. Дан набор строк s_i и строка t. Для каждой строки s_i найдите ее самое левое и самое правое вхождение в t. Время работы: $\mathcal{O}(\sum|s_i| + |t|)$.

8.116. Дан набор строк s_i и строка t. Проверьте, можно ли разрезать строку t на какое-то количество строк таким образом, чтобы каждая строка находилась в множестве s_i. Время работы: $\mathcal{O}(\sum|s_i| + |t|^2)$.

8.117. Есть две прямоугольные таблицы, заполненные буквами: большая и маленькая. Найти число способов вырезать из большой таблицы прямоугольник, равный маленькой таблице.

Неделя 9. Цифровой бор

Устная часть

9.118. Добавьте в Y-fast trie операцию: «найти сумму всех чисел множества, лежащих в диапазоне $[l, r]$». Время работы: $\mathcal{O}(\log w)$.

9.119. Как реализовать сортировку массива целых чисел, используя X-fast trie и Y-fast trie? В каких случаях данное решение будет оптимальным?

9.120. Научитесь искать наибольшую возрастающую подпоследовательность в массиве целых чисел, не превосходящих C по модулю, за $\mathcal{O}(n \log \log C)$.

9.121. Как можно применить X-fast trie и Y-fast trie для ускорения алгоритмов поиска кратчайших путей в графе с целыми неотрицательными весами ребер?

9.122. Как реализовать X-fast trie и Y-fast trie, если битность целых чисел w изначально не известна, не испортив оценки на время работы и память?

9.123. Научитесь искать k-ю порядковую статистику при помощи Y-fast trie за $\mathcal{O}(\log w)$.

9.124. Дан отсортированный массив целых чисел длины n. Научитесь строить по данному массиву Y-fast trie, содержащее все числа массива, за $\mathcal{O}(n)$.

9.125. Дан массив целых чисел. Найдите пару чисел (a_i, a_j), такую что значение $a_i \oplus a_j$ максимально. Время работы: $\mathcal{O}(n \log C)$.

9.126. Решите предыдущую задачу за $\mathcal{O}(n \log \log C)$.

9.127. Дан массив целых чисел. Найдите пару чисел (l, r) $(l \le r)$, такую что значение $a_l \oplus a_{l+1} \oplus \ldots \oplus a_r$ максимально. Время работы: $\mathcal{O}(n \log C)$.

9.128. Задан набор, состоящий из n непустых строк $s_1, s_2, \ldots, s_n$. Во время игры два игрока вместе строят слово, изначально это слово пустое. Игроки ходят по очереди. За свой ход игрок должен дописать в конец слова одну букву так, чтобы полученное слово было префиксом хотя бы одной строки из заданного набора. Проигрывает тот, кто не может сделать ход. Определите, кто выиграет при оптимальной игре обоих игроков. Время работы: $\mathcal{O}(\sum |s_i|)$

Неделя 10. Суффиксный массив

Устная часть

10.129. Дана строка t. Необходимо отвечать на запросы: «найти количество вхождений строки s в t». Время работы на запрос: $\mathcal{O}(|s| + \log|t|)$. Препроцессинг: $\mathcal{O}(|t|\log|t|)$.

10.130. Дана строка t. Необходимо отвечать на запросы: «найти количество вхождений строки s в $k \cdot t$ (строка t, повторенная k раз)». Время работы на запрос: $\mathcal{O}(|s|\log|t|)$. Препроцессинг: $\mathcal{O}(|t|\log|t|)$.

10.131. Найдите количество различных подстрок строки s за $\mathcal{O}(|s|\log|s|)$.

10.132. Найдите сумму длин всех различных подстрок строки s за $\mathcal{O}(|s|\log|s|)$.

10.133. Найдите лексикографически минимальный циклический сдвиг строки s за $\mathcal{O}(|s|\log|s|)$.

10.134. Даны строки s_1 и s_2. Найдите наибольшую общую подстроку строк за $\mathcal{O}(\sum|s_i|\log(\sum|s_i|))$.

10.135. Решите предыдущую задачу для k строк.

10.136. Дана стока s. Найдите такую подстроку t строки s, для которой величина $|t| \cdot c$ максимальна, где c — количество вхождений строки t в строку s. Время работы: $\mathcal{O}(|s|\log|s|)$.

Неделя 11. Корневые оптимизации

Устная часть

11.137. Пусть дана задача о рюкзаке: даны n предметов с весами $w_1, w_2, \ldots, w_n$. Требуется определить, можно ли набрать суммарный вес, в точности равный W. Также известно, что $\sum w_i = S$. Мы уже умеем решать задачу за $\mathcal{O}(n \cdot W)$. Научитесь решать задачу за $\mathcal{O}(S\sqrt{S})$.

11.138. Дано изначально пустое множество строк D. Требуется отвечать на запросы:

- ▷ Добавить строку s в множество D.
- ▷ Удалить строку s из множества D.
- ▷ Для заданной строки s найти количество вхождений строк из множества D в нее. Если некоторая строка p из множества D имеет несколько вхождений в s, необходимо посчитать их все.

Сумма длин строк во всех запросах равна L. Требуется решить задачу за $\mathcal{O}(L\sqrt{L})$.

11.139. Дан неориентированный граф на n вершинах и m ребрах. В каждой вершине записано целое число. Необходимо отвечать на запросы двух типов:

- ▷ Прибавить ко всем соседям вершины v число x
- ▷ Вывести значение, хранящееся в вершине v

Асимптотика: $\mathcal{O}((m+q)\sqrt{m})$.

11.140. Дано дерево на n вершинах. Изначально все вершины не покрашены. Необходимо отвечать на запросы двух типов:

- ▷ Покрасить вершину v.
- ▷ Найти ближайшую к v покрашенную вершину.

Асимптотика: $\mathcal{O}((n+q)\sqrt{n+q})$. Можно сделать препроцессинг.

11.141. Дан массив из n целых чисел. Требуется ответить на q запросов: «найти MEX множества чисел $a_l, a_{l+1}, \ldots, a_r$». MEX множества — это минимальное неотрицательное целое число, не содержащееся в множестве. Время работы: $\mathcal{O}((n+q)\sqrt{n+q})$.

11.142. Решите предыдущую задачу на дереве (в каждой вершине записано целое число, нужно находить MEX на пути).

11.143. Дан массив из n целых чисел и q запросов одного из трех типов:

- ▷ Найти минимум на отрезке $[l, r]$.
- ▷ Развернуть подотрезок массива $[l, r]$.
- ▷ Прибавить ко всем элементам отрезка $[l, r]$ число x.

Время работы: $\mathcal{O}((n+q)\sqrt{n+q})$.