1 Обход в глубину

1.1. Инспектору нужно проверить состояние дорог в городе, для этого он хочет проехать по каждой дороге в каждую сторону (все дороги двусторонние). Постройте кратчайший путь. $O(n+m)$.

1.2. Правда ли, что если в ориентированном графе существует путь из u в v, и время выхода из вершины u больше, чем время выхода из вершины v, то u является предком v в дереве обхода в глубину?

1.3. Может ли в лесе обхода в глубину получиться дерево из одной вершины, если у этой вершины есть и входящие и исходящие ребра?

1.4. Удалить из орграфа минимальное число ребер так, чтобы он стал ациклическим.

1.5. Правда ли, что для топологической сортировки орграфа можно сортировать не по убыванию времени выхода, а наоборот, по возрастанию времени входа?

1.6. Запустим алгоритм построения топологической сортировки на орграфе с циклами. Правда ли, что он минимизирует число ребер, ведущих справа налево?

1.7. Проверьте, что в данном орграфе есть только один способ построить топологическую сортировку. $O(n+m)$.

1.8. Найдите для каждой вершины орграфа самую левую и самую правую из возможных позиций в топологической сортировке. $O(nm)$.

1.9. Постройте лексографически минимальную топологическую сортировку (если представлять топологическую сортировку как последовательность номеров вершин). $O((n+m)\log n)$.

1.10. Постройте в заданном ацикличном орграфе гамильтонов путь, или скажите, что его нет. $O(n+m)$.

1.11. Найти в ациклическом взвешенном орграфе кратчайший путь из s в t. $O(n+m)$.

1.12. Модифицируйте алгоритм поиска цикла, чтобы он искал любой простой цикл в заданном неориентированном графе.

1.13. Дан ациклический орграф, на каждом ребре написана буква. Найти лексикографически минимальный путь из s в t. $O(n+m)$.

1.14. Транзитивным замыканием графа называется такой граф, в котором ребро (u, v) есть тогда и только тогда, когда в исходном графе есть путь из u в v. Построить транзитивное замыкание. $O(nm)$.

1.15. Транзитивным остовом графа называется минимальный граф, транзитивное замыкание которого совпадает с транзитивным замыканием исходного графа. Построить транзитивный остов. $O(nm)$.

2 Компоненты сильной связности

2.1. Дан ориентированный граф. Какое минимальное число ребер нужно добавить в него, чтобы он стал сильно связным?

2.2. Дан ориентированный граф. Постройте граф, с таким же множеством вершин и минимальным числом ребер, чтобы его конденсация совпадала с конденсацией данного графа.

2.3. В городе n перекрестков, соединенных m односторонними дорогами. Нужно установить в городе несколько пунктов быстрого реагирования. Установить пункт на перекрестке i стоит c_i рублей. Сколько денег нужно потратить, чтобы до каждого перекрестка можно было бы доехать хотя бы из одного построенного пункта?

2.4. Для каждой вершины графа найти вершину с минимальным номером, достижимую из нее.

2.5. Турниром называется ориентированный граф, в котором каждая пара вершин соединена ребром (в одну или другую сторону). Докажите, что в сильно связном турнире есть гамильтонов цикл.

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

2.7. Дано число n и и m пар (a_i, b_i). Нужно построить ориентированный граф на n вершинах с минимальным числом ребер, в котором для каждой пары существует путь из a_i в b_i.

3 2-SAT и смежные задачи

3.1. Найти решение 2-SAT, в котором число переменных, равных 1, минимально. Шутка, это NP-полная задача. Но если для каждой скобки $(X|Y)$ есть парная скобка $(!X|!Y)$ (а для $(!X|Y)$ соответственно $(X|!Y)$), то задачу можно решить. Как?

3.2. Найти решение 2-SAT, в котором ответ лексикографически минимален (то есть последовательность из значений x_1, x_2, ..., x_n лексографически минимальна из всех последовательностей, на которых выполняется формула). (Это же явно сложнее, чем просто минимизировать число единиц, наверняка тоже NP-полная, да?)

3.3. Найдите следующее в лексикографическом порядке решение 2-SAT.

3.4. Дан граф, нужно раскрасить его вершины в три цвета так, чтобы никакие две вершины, соединенные ребром не были одного цвета. Это тоже NP-полная задача, но мы сделаем ее даже сложнее: для каждой вершины задан цвет, в который ее запрещено красить.

3.5. Есть n переменных, каждая из которых может принимать значения от 1 до k и m следствий вида $x_i = a \Rightarrow x_j = b$. Найдите значения, которые могут принимать переменные.

3.6. Есть n различных целых чисел: p_1, p_2, ..., p_n и константы a и b. Нужно разделить все числа на два множества, A и B, чтобы выполнялись условия: Если число x принадлежит множеству A, то число $a - x$ также должно принадлежать множеству A. Если число x принадлежит множеству B, то число $b - x$ также должно принадлежать множеству B.

3.7. Есть n городов, образующих кольцо — пары городов с номерами i и $i + 1$, а так же n и 1 соединены дорогами. Нужно построить m новых дорог, для каждой дороги известно какие города она должна соединять. Каждая дорога будет представлять собой непрерывную линию, находящуюся внутри, либо снаружи кольца. Дороги не должны иметь общих точек (кроме концов). Можно ли построить дороги? Если возможно, то какие дороги должны находиться внутри кольца, а какие снаружи?

3.8. Есть n булевых переменных и m утверждений вида $x_i = x_j$ или $x_i \neq x_j$. Найдите решение, в котором разница между числом переменных, равных 0 и числом переменных, равных 1, минимальна.

3.9. Дан неориентированный граф из n вершин и m рёбер. Каждое из рёбер графа изначально покрашено либо в красный, либо в синий цвет. За один ход разрешается выбрать произвольную вершину, и изменить цвета всех инцидентных ей рёбер, то есть все красные рёбра, заканчивающиеся в этой вершине, становятся синими, и наоборот, все синие становятся красными.

Найдите кратчайшую последовательность ходов, после выполнения которой все рёбра графа будут покрашены в один цвет.

4 Мосты, точки сочленения

4.1. Дан неориентированный граф. Сколько ребер нужно добавить в него, чтобы он стал реберно двусвязным?

4.2. Дан неориентированный связный граф. Ориентировать его ребра так, чтобы получился сильно связный граф.

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

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

4.5. Дан граф. После предподсчета за $O(E + V \log V)$ нужно за $O(\log V)$ отвечать на запросы: для данных вершин u и v, сколько есть ребер, по которым мы обязательно пройдем, если будем идти из u в v.

4.6. Дан граф. После предподсчета за $O(E + V \log V)$ нужно за $O(\log V)$ отвечать на запросы: для данных вершин u и v, сколько есть вершин, по которым мы обязательно пройдем, если будем идти из u в v.

4.7. Дан граф, на каждом ребре написано число. После предподсчета за $O(E + V \log V)$ нужно за $O(\log V)$ отвечать на запросы: для данных вершин u и v, какое максимальное число можно встретить на реберно простом пути из u в v.

4.8. Дан граф, на каждой вершине написано число. После предподсчета за $O(E + V \log V)$ нужно за $O(\log V)$ отвечать на запросы: для данных вершин u и v, какое максимальное число можно встретить на вершинно простом пути из u в v.

5 Эйлеровы циклы

5.1. Дан связный граф. Добавьте в него минимальное число ребер, чтобы он стал эйлеровым. а) для неориентированного графа, б) для ориентированного графа.

5.2. То же самое, но граф не связный.

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

5.4. Дан неориентированный граф. Найдите в нем реберно простой цикл нечетной длины.

5.5. Дан ориентированный граф. Построить минимальное количество реберно простых путей, так чтобы каждое ребро было ровно в одном пути.

5.6. Дан неориентированный граф, в котором $2k$ вершин нечетной степени. Разбейте эти вершины на k пар и соедините вершины в каждой паре путем так, чтобы полученные пути не имели общих ребер.

6 Еще задачи на DFS и все такое

6.1. Неориентированный граф называется вершинным кактусом, если каждая вершина содержится не более чем в одном простом цикле. Проверить, что данный граф является вершинным кактусом. Время $O(m)$.

6.2. Неориентированный граф называется реберным кактусом, если каждое ребро содержится не более чем в одном простом цикле. Проверить, что данный граф является реберным кактусом. Время $O(m)$.

6.3. Дан вершинный кактус, нужно быстро отвечать на запросы вида: найти число различных реберно простых путей из u в v. Предподсчет $O(n \log n)$, запросы за $O(\log n)$.

6.4. Дан реберный кактус. Каждое ребро uv удаляется с вероятностью p_{uv}. Найти вероятность того, что граф останется связным после удаления ребер. Время $O(m)$.

6.5. Дан неориентированный граф. Нужно выбрать в нем две вершины, между которыми есть три вершинно непересекающихся пути (кроме концов), или сказать, что это сделать нельзя. Время $O(m \log n)$.

6.6. Дан неориентированный граф, в нем выделены некоторые вершины. Известно, что эти вершины не соединены путями друг с другом. Добавить в граф максимальное число ребер, чтобы это свойство сохранялось (нельзя делать параллельные ребра).

6.7. Дан неориентированный граф, на каждой вершине написана буква A, B или C. Можно поставить в вершину графа фишку и двигать ее вдоль ребер, при этом из вершины A можно перейти только в B, из B только в C, из C только в A. Определите, какое максимальное число ходов можно сделать.

6.8. Есть набор костяшек домино. Можно ли их все выложить в линию по правилам домино?

6.9. Был полный граф из n вершин. Из него удалили m ребер. Выделить компоненты связности за $O(n + m)$.

6.10. Дан ориентированный граф. Найти все такие вершины v, что для любой вершины u существует путь либо из u в v, либо из v в u. Время $O(m \log n)$.

6.11. Дан неориентированный граф. Найти все такие ребра, при удалении которых граф становится двудольным. Время $O(m \log n)$.

6.12. 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 $O(m + n)$.

6.13. Есть булева функция в виде 2-КНФ для $n+m$ переменных. Проверить, что для любых значений $x_1, x_2, \ldots, x_n$ существуют значения $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$, при которых функция выполняется.

7 Минимальное остовное дерево

7.1. Рассмотрим путь из v в u, в котором вес максимального ребра минимален. Покажите, что можно построить путь с таким же максимумом по ребрам минимального остовного дерева.

7.2. Рассмотрим произвольный цикл в графе. Покажите, что максимальное ребро на этом пути не лежит в минимальном остовном дереве.

7.3. На болоте находятся n кочек. Лягушонку нужно допрыгать от кочки 1 до кочки n. Он маленький и ему тяжело прыгать на большое расстояние. Найдите минимальное d такое, что есть путь, в котором все прыжки не больше d. $O(n^2)$.

7.4. Вам нужно передать с одного компьютера на другой n файлов. Каждый файл — это битовый вектор размера m. Файл можно переслать либо просто как последовательность битов, либо как diff с другим файлом, уже пересланным ранее. В первом случае нужно будет передать m бит, во втором — $A + B \times d$ бит, где d — число различающихся битов, A и B — некоторые константы. Найдите минимальное число битов, которое нужно передать.

7.5. В стране n городов. Однажды в ней решили построить $n-1$ дорогу, чтобы связать все города. Однако что-то пошло не так, и получилось, что эти $n-1$ дорог образуют не дерево, а k компонент связности. Нужно перестроить $k-1$ дорогу (то есть убрать $k-1$ и построить другие $k-1$), так, чтобы получилось дерево. Из всех таких способов нужно выбрать такой, чтобы суммарная длина дорог была минимальна.

7.6. Есть n городов. Можно соединить два города дорогой, потратив $A \times len$ денег, где len — длина дороги, а можно построить в городе аэропорт, потратив B денег. Нужно за минимальное число денег соединить все города (чтобы от каждого до каждого можно было добраться с помощью дорог и самолетов).

7.7. Для каждого ребра проверить а) правда ли, что есть минимальное остовное дерево, которое его содержит, б) правда ли, что есть минимальное остовное дерево, которое его не содержит.

7.8. Найти число способов постронить минимальное остовное дерево в данном графе.

7.9. Дано значение d, требуется найти минимальное остовное дерево, в котором степень вершины номер 1 не превосходят d.

7.10. Пусть в задаче о минимальном остовном дереве добавлено следующее ограничение: для каждой вершины v задано значение $d[v]$. Требуется найти минимальное остовное дерево, в котором степени вершин не превосходят соответствующих значений $d[v]$. Покажите, что эта задача не проще, чем задача о нахождении минимального гамильтонова пути (которая, как известно, NP-полна).

8 Ориентированное остовное дерево

8.1. Приведите пример графа, в котором минимальное ориентированное остовное дерево не содержит минимальное ребро графа, и при этом есть какое-то ориентированное остовное дерево, которое его содержит.

8.2. Приведите пример графа, в котором минимальное ориентированное остовное дерево содержит максимальное ребро графа, и при этом есть какое-то ориентированное остовное дерево, которое его не содержит.

8.3. Пусть нужно построить минимальное ориентированное остовное дерево, но корень дерева не фиксирован и может быть любой вершиной. Решите эту задачу за то же время, что и задачу с фиксированным корнем.

8.4. Пусть нужно построить минимальный ориентированный лес (то есть множество ребер в котором нет циклов и в каждую вершину входит не более одного ребра). Как модифицировать алгоритм, чтобы он решал эту задачу?

8.5. Постройте минимальное ориентированное остовное дерево в ациклическом графе за линейное время.

8.6. Покажите, что следующий жадный алгоритм не работает: начнем с дерева из одной вершины, на каждой итерации будем добавлять в дерево минимальное ребро, один из концов которого уже в дереве, а другой — еще нет.

9 Обход в ширину

9.1. Дан ориентированный граф. Найдите все ребра, для которых существует кратчайший путь из s в t, который через него проходит. Время $O(m)$.

9.2. Дан ориентированный граф. Найдите минимальное число ребер, которое нужно развернуть, чтобы был путь из s в t. Время $O(m)$.

9.3. Дан ориентированный граф. Разрешается развернуть не более одного ребра, требуется, чтобы расстояние от s до t было минимально. Время $O(m)$.

9.4. Дан ориентированный граф, на каждом ребре которого написана буква. Рассмотрим все кратчайшие (по числу ребер) пути из вершины s в вершину t. Для каждого пути выпишем строку из соответствующих букв. Найдите путь, для которого эта строка будет лексикографически минимальной. Время $O(m)$.

9.5. Калькулятор умеет делать две операции: `a = (a + 3) % M` и `a = (a * 4) % M`. За сколько операций можно получить из числа a число b? Время $O(M)$.

9.6. Есть n типов монет, номинал каждой не более M, число монет каждого типа не ограничено. Можно ли этими монетами набрать сумму S? Время $O(M \cdot n)$ (обратите внимание, что S может быть большим).

9.7. Дан граф, веса ребер равны 0 или 1. Найти кратчайший путь. Время $O(m)$.

9.8. Дан граф, веса ребер — целые числа от 1 до k. Найти кратчайший путь. Время $O(m + kn)$.

10 Алгоритм дейкстры

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

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

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

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

10.5. Приведите пример графа с отрицательными рёбрами, на котором алгоритм Дейкстры работает неверно.

10.6. Приведите пример графа, на котором алгоритм Дейкстры делает $\Omega(n^2)$ успешных релаксаций.

10.7. Индиана Джонс хочет выбраться из лабиринта, который представляет собой граф. Для каждого ребра известно время, за которое его можно пробежать, в одной из вершин находится герой, в какой-то другой — выход. Чтобы было динамичнее, лабиринт постепенно разрушается. Для каждого ребра известно время, в которое он разрушится. Найдите кратчайший путь до выхода или скажите, что все плохо.

11 Кратчайшие пути и не только

11.1. Дан граф, на каждом ребре написано число. За один ход можно пройти несколько ребер, сумма чисел на которых не превышает D. Дойти от s до t за минимальное число ходов. $O(m \log n)$.

11.2. Есть сеть железных дорог, про каждую известно, сколько можно получить денег, если сдать ее на металлалом, нужно получить как можно больше, но чтобы города 1 и 2 были связаны.

11.3. Петя перепутал и написал алгоритм Флойда так:
`for i: for j: for k: d[i][j] = min(d[i][j], d[i][k] + d[k][j])`.
Постройте тест, на котором получившийся алгоритм работает неверно.

11.4. Как найти все вершины, вес пути до которых может быть сколь угодно малым?

11.5. Приведите пример графа без отрицательных циклов, на котором алгоритм Форда–Беллмана делает $\Omega(n^3)$ успешных релаксаций.

11.6. Дан взвешенный неориентированный связанный граф, $m - n \leq 20$. Вам нужно отвечать на запросы: найти кратчайшее расстояние между двумя заданными вершинами. Предподсчет за $O(m \log n)$, ответ на запрос за $O(1)$.

11.7. Модифицировать алгоритм Дейкстры, чтобы он работал за $O(M + m)$, если все кратчайшие пути не больше M.

11.8. Дан граф, изначально веса ребер равны нулю. Есть операция: заданному множеству ребер увеличить вес на 1. Нужно после каждого изменения пересчитать расстояния от s до всех вершин за $O(n + m)$.

11.9. Дан взвешенный граф. Удалить максимальное число ребер, при условии что расстояние от s до t должно быть не больше d.

11.10. Дано взвешенное дерево. Найдите расстояние между каждой парой вершин. Время $O(n^2)$.

11.11. Сеть дорог представляет собой взвешенное дерево. В каждом городе продаются машины, в городе i можно купить машину, которая едет со скоростью v_i. За какое минимальное время можно доехать из города A в город B? Время $O(n^2)$.

11.12. Есть n планет и m телепортов. С помощью телепорта i можно попасть с планеты v_i на любую из планет с номерами от l_i до r_i. Найти кратчайший путь от планеты 1 до планеты n. Время $O((n + m) \log n)$.

11.13. Дан взвешенный неориентированный граф, веса ребер неотрицательные. Выделены три вершины. Выбрать подграф минимального веса, в котором эти вершины лежат в одной компоненте связности. Время $O(m \log n)$.

11.14. То же задание для четырех выделенных вершин. Время $O(m \log n)$.

11.15. 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?

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

11.17. Петя и Вася играют в игру. Дан граф, в котором выделена вершина s. Петя выбирает вершину t и строит какой-то реберно-простой путь из s в t. После этого Вася должен построить еще один реберно-простой путь из s в t, не пересекающийся по ребрам с путем Пети. Если он сможет это сделать, он выиграл, если нет — то выиграл Петя. Кто выиграет при правильной игре? (подсказка: посмотрите, кто выигрывает если граф Эйлеров)

11.18. Дано n чисел x_i. Постройте ориентированный ацикличый граф из n вершин, в котором из вершины i достижимо ровно x_i вершин (или скажите, что такого нет).

12 Игры на графах

12.1. Пусть есть две обычные игры на графе, но, в отличие от обычной суммы игр, на каждом ходу игрок делает ход в обеих играх. Если хотя бы в одной фишка в тупике, то игрок проигрывает. Кто выиграет при правильной игре? Время $O(n+m)$.

12.2. Рассмотрим обычную игру на графе, но первый игрок хочет, чтобы игра закончилась (при этом ему не важно, кто выиграет), а второй — чтобы длилась вечно. Кто добьется своего?

12.3. Рассмотрим обычную игру на графе, но каждый игрок торопится домой, поэтому если у него нет возможности выиграть, то он хочет проиграть (а не ходить бесконечно по циклу). Что не так в этой игре?

12.4. Пусть первый игрок торопится домой, как в предыдущей задаче, а второй — наоборот, хочет, если возможно, ходить вечно по циклу, а если нельзя, то выиграть (завести противника в тупик). Эта игра лучше предыдущей? Как узнать результат такой игры?

12.5. Рассмотрим асимметричные игры на графе (ребра двух цветов, первый ходит только по первым, второй только по вторым). Пусть мы хотим проанализировать сумму двух таких игр. Почему нельзя просто свести каждую из игр по отдельности к обычным играм и использовать функцию Гранди на полученных играх?

12.6. Рассмотрим игру в поддавки (когда игрок выигрывает в тупике). Пусть мы хотим проанализировать сумму двух таких игр. Почему нельзя просто свести каждую из игр по отдельности к обычным играм и использовать функцию Гранди на полученных играх?

12.7. Есть полоска из n клеток. Игроки по очереди делают ходы. Ход заключается в том, чтобы поставить доминошку на две соседние свободные клетки. Если игрок не может поставить доминошку, он проигрывает. Кто выиграет? Время $O(n^2)$.

12.8. Есть полоска из n клеток. Игроки по очереди делают ходы. Ход заключается в том, чтобы поставить фишку на свободную клетку. При этом нельзя ставить фишку на клетку, если в соседней с ней уже стоит фишка. Кто не может сделать ход, тот проиграл. Кто выиграет? Время $O(n^2)$.

12.9. Есть полоска из n клеток. Игроки по очереди делают ходы. Ход заключается в том, чтобы поставить фишку на свободную клетку. Если после хода игрока три фишки стоят подряд, то он выигрывает. Кто выиграет? Время $O(n^2)$.

12.10. Разберите игру Ним.

12.11. Разберите игру Хакенбуш.

12.12. Ним в поддавки: кто взял последний камень тот проиграл.

13 Дополнительные задачи на графы

13.1. Даны m троек (a_i, b_i, x_i) и m троек (c_i, d_i, y_i). Построим граф из $n \times n$ вершин, каждая вершина которого задается парой чисел (i, j) со следующими ребрами: для каждого $i = 1..m$ и каждого $j = 1..n$ есть ребро $(a_i, j) - (b_i, j)$ веса x_i и ребро ребро $(j, c_i) - (j, d_i)$ веса y_i. Найти вес минимального остовного дерева в этом графе. $O((n+m)\log(n+m))$.

13.2. Даны n чисел a_i. Построим полный граф из n вершин, вес ребра $u - v$ равен $|a_u - a_v|$. Найдите вес минимального остовного дерева в этом графе. $O(n \log n)$.

13.3. Дан полный граф из n вершин. Подобрать такие числа a_i, чтобы вес ребра $u - v$ был равен $|a_u - a_v|$.

13.4. Дан ориентированный граф, некоторые вершины помечены. Найти путь (не обязательно простой), проходящий через все помеченные вершины.

13.5. Игра устроена так: есть прямоугольное поле, некоторые клетки свободны, некоторые заполнены стеной. В некоторых свободных клетках находятся звезды. Игрок находится в какой-то свободной клетке. За ход можно направить его в одну из четырех сторон, тогда он будет двигаться до ближайшего препятствия, собирая все звезды на пути. Можно ли собрать все звезды?

14 Строки: хеши

В следующих заданиях предположим, что коллизий хеш-функции не бывает (то есть, если хеш-функции строк равны, то можно считать, что строки равны)

14.1. Дана строка. Надо отвечать на запросы `lcp(i, j)`: длина наибольшего общего префикса строк $s[i..n]$ и $s[j..n]$. $O(\log n)$ на запрос.

14.2. Дана строка. Надо отвечать на запросы: является ли заданная подстрока палиндромом? $O(\log n)$ на запрос.

14.3. Дана строка. Надо отвечать на запросы: являются ли две заданные подстроки анаграммами? $O(\log n)$ на запрос.

14.4. Число $p < |s|$ является периодом строки s, если для любого i $s[i] = s[i+p]$. Дана строка. Найти все ее периоды. $O(n)$.

14.5. Даны две строки. Найдите их наибольшую общую подстроку. $O(n \log n)$.

14.6. Дана строка. Найти максимальную по длине строку, которая является ее префиксом, суффиксом, а так же встречается в середине строки. $O(n \log n)$.

14.7. Дана строка. Обрабатывать два вида запросов: 1) изменить один символ 2) проверить что две подстроки равны. Оба запроса за $O(\log n)$.

15 Строки: префикс-функция, КМП

Задания этого раздела надо решать без хешей

15.1. Дана строка. Найти все ее периоды. $O(n)$.

15.2. Дана строка. Найти все ее префиксы, которые являются палиндромами. $O(n)$.

15.3. Дана строка. Для каждого ее префикса посчитайте, сколько раз он встречается в строке как подстрока. $O(n)$.

15.4. Дана строка. Для каждого ее префикса посчитайте, сколько есть строк, которые являются его префиксом и суффиксом. $O(n)$.

15.5. Постройте строку с заданной префикс-функцией. $O(n)$.

15.6. Две последовательности чисел назовем эквивалентными, если одну можно получить из другой добавлением одного числа ко всем элементам (Например, последовательности $(3, 6, 4)$ и $(7, 10, 8)$). Даны массивы A и B, найдите в массиве A все подмассивы, эквивалентные B. $O(n)$.

15.7. 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). Даны строки A и B, найдите в строкие A все подстроки, эквивалентные B. $O(n)$.

16 Строки: Z-функция

16.1. Дана строка. Найти максимальную по длине строку, которая является ее префиксом, суффиксом, а так же встречается в середине строки. $O(n)$.

16.2. Постройте строку с заданной Z-функцией. $O(n)$.

16.3. Постройте Z-функцию по префикс-функции, не восстанавливая строку. $O(n)$.

16.4. Постройте префикс-функцию по Z-функции, не восстанавливая строку. $O(n)$.

16.5. Даны строки S и T, найдите в S подстроки, отличающиеся от T одной буквой. $O(n)$.

16.6. Дана строка S. Найдите число суффиксов S, которые лексикографически меньше S $O(n)$.

16.7. 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 n найдите самое короткое представление в таком виде. Время $O(n^3)$.

17 Строки: Бор

17.1. Дан набор строк s_i. Отсортировать их в лекскографическом порядке за время $O(\sum |s_i|)$ (константный алфавит).

17.2. Дан набор строк s_i. Найти самую короткую строку, которая не является префиксом никакой из них.

17.3. Дан набор строк s_i. Отвечать на запросы: по данным (i, j) найти длину наибольшего общего префикса строк s_i и s_j.

17.4. 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?

17.5. Придумайте, как хранить сжатый бор, потратив $O(n)$ дополнительной памяти (вместо $O(\sum |s_i|)$).

17.6. Покажите, что глубина сжатого бора $O(\sqrt{\sum |s_i|})$.

17.7. Дан набор строк s_i. Построить бор с минимальным числом вершин, в котором все заданные строки встречаются как подстроки.

17.8. Предположим, что вы ищете в боре только те строки, которые в нем точно есть. Придумайте, как хранить такой бор, потратив $O(n)$ памяти (вместо $O(\sum |s_i|)$).

17.9. Предположим, что у хешей нет коллизий. Придумайте, как хранить бор, потратив $O(n)$ памяти (вместо $O(\sum |s_i|)$).

18 Алгоритм Ахо-Корасик

18.1. Дан набор строк s_i. Проверьте, существует ли строка длины n, не содержащая ни одну из s_i как подстроку. $O(n \cdot \sum len(s_i))$.

18.2. Дан набор строк s_i. Проверьте, существует ли строка бесконечной длины, не содержащая ни одну из s_i как подстроку. $O(\sum len(s_i))$.

18.3. Дан набор строк s_i и строка t. Нужно вырезать из строки t максимальное количество непересекающихся подстрок, так чтобы каждая из них была в множестве s_i (можно вырезать несколько одинаковых строк). $O(\sum len(s_i) + len(t))$.

18.4. Даны строки s_1, s_2 и t. Найти непересекающиеся (не имеющие общих символов) вхождения строк s_1 и s_2 в t. $O(\sum len(s_i) + len(t))$.

18.5. Даны строки s_1, s_2 и t. Найти пересекающиеся (имеющие общий символ) вхождения строк s_1 и s_2 в t. $O(\sum len(s_i) + len(t))$.

18.6. Дан набор строк s_i и строка t. Для каждой из s_i найти самое левое и самое правое вхождение в t. $O(\sum len(s_i) + len(t))$.

18.7. Дан набор строк s_i и строка t. Проверить, можно ли разбить строку t на куски так, чтобы каждый был из множества s_i. $O(\sum len(s_i) + len(t)^2)$.

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

18.9. Дан набор из n строк s_i, отсортированных в лексикографическом порядке, кроме того для каждой пары соседних строк i и $i+1$, известна длина их наибольшего общего префикса lcp_i. Постройте сжатый бор для данного набора строк за $O(n)$.

19 Суффиксный массив

19.1. Отвечать на запросы: число вхождений строки s в t. Предподсчет за $O(|t| \log |t|)$, ответ за $O(|s| + \log |t|)$.

19.2. Отвечать на запросы: входит ли строка s в строку $k \cdot t$ (k повторений строки t). Предподсчет за $O(|t| \log |t|)$, ответ за $O(|s| \log |t|)$.

19.3. Посчитать число различных подстрок в строке t за $O(|t| \log |t|)$.

19.4. Посчитать сумму длин всех различных подстрок в строке t за $O(|t| \log |t|)$.

19.5. Найти лексикографически минимальный циклический сдвиг строки t за $O(|t| \log |t|)$.

19.6. Дана строка t. Найти самую длинную строку, которая встречается в t не менее k раз за $O(|t| \log |t|)$.

19.7. Даны две строки t_1 и t_2. Найти самую длинную общую подстроку. Время $O(\sum |t_i| \log(\sum |t_i|))$.

19.8. То же для k строк.

19.9. Дана строка t. Найти самую короткую строку s, такую, что t можно «покрыть» строками s. То есть, для любого символа строки t есть подстрока t, равная s, содержащая этот символ. Например: для $t =$ "*ababbababbabababbabababbababbaba*" минимальная строка $s =$ "*ababbaba*".

```
.ababbababbabababbabababbababbaba
.ababbaba
.     ababbaba
```

```
.              ababbaba
.                     ababbaba
.                          ababbaba
```

20 Суффиксное дерево

Да, вы уже где-то видели эти задачи... Но теперь их надо решить с помощью суффиксного дерева.

20.1. Отвечать на запросы: число вхождений строки s в t. Предподсчет за $O(|t|)$, ответ за $O(|s|)$.

20.2. Отвечать на запросы: входит ли строка s в строку $k \cdot t$ (k повторений строки t). Предподсчет за $O(|t|)$, ответ за $O(|s|)$.

20.3. Отвечать на запросы: сколько раз подстрока $s[l..r]$ встречается в строке s? Предподсчет за $O(|s|)$, ответ за $O(\log |s|)$.

20.4. Найти число различных подстрок строки t за $O(|t|)$.

20.5. Посчитать сумму длин всех различных подстрок строки t за $O(|t|)$.

20.6. Найти лексикографически минимальный циклический сдвиг строки t за $O(|t|)$.

20.7. Дана строка t. Найти самую длинную строку, которая встречается в t не менее k раз за $O(|t|)$.

20.8. Даны две строки t_1 и t_2. Найти самую длинную общую подстроку. Время $O(\sum |t_i|)$.

20.9. То же для k строк.

20.10. Даны строки s и t. Отвечать на запросы $lcp(i, j)$: максимальное l, такое что $s[i..i+l] = t[j..j+l]$.

20.11. Дана строка t. Найти самую длинную подстроку t, которая является палиндромом. $O(|t|)$.