АиСД 2023. Третий семестр

Задания на практики M3232-M3233

⟨Версия от 23 декабря 2023 г.⟩

Темы

Неделя 1. Поиск в глубину, топологическая сортировка

Практика

1.1. В произвольном неориентированном графе за время $\mathcal{O}(n+m)$

   (a) найдите для каждой вершины количество достижимых из нее
   (b) найдите для каждой вершины минимальную по номеру вершину, достижимую из нее
   (c) проверьте граф на двудольность
   (d) проверьте существование нечетного цикла
   (e) если на каждом ребре написана буква, найдите путь из s в t, задающий лексикографически минимальную строку (гарантируется, что ни из какой вершины не выходит два ребра с одной буквой)

1.2. В произвольном ориентированном графе

   (a) найдите для каждой вершины количество достижимых из нее за время $\mathcal{O}(n+m)$
   (b) найдите для каждой вершины минимальную по номеру вершину, достижимую из нее, за время $\mathcal{O}(n+m)$
   (c) проверьте единственность топологической сортировки за время $\mathcal{O}(n+m)$
   (d) постройте транзитивное замыкание за время $\mathcal{O}(nm)$

1.3. В произвольном DAG'е за время $\mathcal{O}(n+m)$

   (a) проверьте существование гамильтонова пути
   (b) найдите количество кратчайших путей из s в t

1.4. Для ориентированного графа найдите наименьшее множество вершин, из которых достижимы все остальные, за время $\mathcal{O}(n+m)$.

1.5. На ориентированном графе запустили обход в глубину, получили лес (множество деревьев) обхода. Может ли вершина с $\texttt{deg}_{in}(v) > 0$ и $\texttt{deg}_{out} > 0$ образовывать в этом лесу дерево из одной вершины?

1.6. За время $\mathcal{O}(n+m)$ определите минимальный по размеру набор ребер, который надо удалить из ориентированного графа, чтобы он стал ациклическим.

1.7. Можно ли для топологической сортировки использовать только время входа вместо времени выхода?

1.8. Вам дан фрагмент кода

```
fun dfs(v: Int) {
    visited++
    for (u in graph[v]) {
        if (visited < n) dfs(u)
    }
}

visited = 0
for (v in 0..n - 1) {
    if (visited < n) dfs(v)
}
```

Работает ли он 1. за время $\mathcal{O}(n+m)$ и 2. как корректный `dfs`

(a) на DAG'е?

(b) на произвольном ориентированном графе

Идейные задачи

1.9. Проверьте в графе существование четного и нечетного пути из вершины s во все остальные вершины за время $\mathcal{O}(n+m)$.

1.10. В произвольном

(a) DAG'е

(b) ориентированном графе

некоторые вершины помечены, а ребра покрашены в черный или белый цвета. За время $\mathcal{O}(n+m)$ определите для каждой вершины, существует ли из нее путь до помеченной, чередующий цвета ребер.

1.11. Есть n гирек разных весов. Известна последовательность результатов их взвешиваний (про две гирьки становится известно, какая из них тяжелее). Определите, корректны ли данные результаты, или найдите первое взвешивание, после которого они становятся некорректными, за время $\mathcal{O}((n+m)\log n)$.

1.12. В произвольном ориентированном графе за время $\mathcal{O}((n+m)\log n)$

(a) постройте лексикографически минимальную топологическую сортировку

(b) удалите не более k ребер, чтобы лексикографически минимальная топологическая перестановка стала как можно меньше

1.13. Есть n человек, у каждого среди этого множества есть не более трех врагов. Отношение вражды симметрично. За время $\mathcal{O}(n)$ разбейте людей на две группы так, чтобы у каждого человека вместе с ним в группе было не более одного его врага.

1.14. Найдите количество «треугольников» в графе за время

(a) $\mathcal{O}(n^3)$

(b) $\mathcal{O}(nm)$

1.15. **Маски**. Представьте, что список соседей каждой вершины хранится в `std::bitset` (как маска множества).

(a) Найдите количество треугольников в графе за время $\mathcal{O}\left(\frac{nm}{\texttt{word\_size}}\right)$.

(b) Найдите матрицу достижимости за время $\mathcal{O}\left(\frac{nm}{\texttt{word\_size}}\right)$.

Неделя 2. Компоненты сильной связности, `2-SAT`

Практика

2.1. Какое

(a) минимальное

(b) максимальное

количество ребер может быть в ориентированном графе на n вершинах с k компонентами сильной связности?

2.2. Приведите контрпример для каждой из следующих версий алгоритма поиска компонент сильной связности.

Пояснение: Здесь `tin` и `tout` – указание на то, сортировать вершины по времени входа или времени выхода, $\uparrow$ и $\downarrow$ – в порядке возрастания или убывания, а G и $G^{\mathtt{rev}}$ – делать `dfs` по прямым или обратным ребрам.

(a) `dfs2` в порядке `tin` $\uparrow$ по G

(b) `dfs2` в порядке `tout` $\uparrow$ по G

(c) `dfs2` в порядке `tin` $\downarrow$ по G

(d) `dfs2` в порядке `tout` $\downarrow$ по G

(e) `dfs2` в порядке `tin` $\uparrow$ по $G^{\mathtt{rev}}$

(f) `dfs2` в порядке `tout` $\uparrow$ по $G^{\mathtt{rev}}$

(g) `dfs2` в порядке `tin` $\downarrow$ по $G^{\mathtt{rev}}$

(-) ~~`dfs2` в порядке `tout` $\downarrow$ по $G^{\mathtt{rev}}$~~

2.3. Дан ориентированный граф на n вершинах и m ребрах. За время $\mathcal{O}(n+m)$ определите, какое минимальное количество ребер надо в него добавить, чтобы он стал сильно связным.

2.4. По данному ориентированному графу постройте за время $\mathcal{O}(n+m)$ другой граф с тем же множеством вершин и минимальным числом ребер, чтобы его граф конденсации совпадал с конденсацией исходного графа.

2.5. **Быстрое реагирование**.

(a) Выделите в ориентированном графе за время $\mathcal{O}(n+m)$ наименьшее множество вершин, из которых достижимы все оставшиеся.

(b) Та же задача, но теперь выбрать вершину номер i стоит c_i денег. Требуется выбрать самое дешевое множество вершин, из которых достижимы все оставшиеся.

2.6. Для каждой вершины ориентированного графа найдите вершину с минимальным номером, достижимую из нее, за время $\mathcal{O}(n+m)$.

2.7. Верны ли следующие утверждения? Приведите доказательство или контрпример.

(a) в графе конденсации ребра ведут от компоненты с большим номером в компоненту с меньшим номером

(b) если $a < b < c$, то компонента номер b находится на пути между компонентами с номерами a и c

(c) вершина исходного графа номер 0 окажется в КСС с максимальным номером (если `dfs` перебирает вершины в порядке возрастания номеров)

> **Def.** *Турниром* назвается ориентированный граф, в котором между каждой парой вершин проведено ребро ровно в одну сторону.

2.8. Докажите, что если граф турнира сильно связен, то в нем существует гамильтонов цикл.

2.9. Докажите, что если в графе турнира для каждой вершины $\texttt{deg}_{in}(v) = \texttt{deg}_{out}(v)$, то этот граф сильно связен.

2.10. Постройте ориентированный граф на n вершинах с минимальным числом ребер, в котором для каждой из m заданных пар вершин (u_i, v_i) существует путь из u_i и v_i.

`2-SAT`

> Во всех заданиях этого блока, если не указана асимптотика решения, подразумевается линейное от размера входных данных время.

2.11. Дана 2-КНФ формула, в которой для каждого клоза $(x \vee y)$ есть «парный» $(\neg x \vee \neg y)$ (и $(x \vee \neg y)$ для $(\neg x \vee y)$, соответственно). Найдите решение `2-SAT` для нее, в котором максимальное количество переменных принимает значение 1.

2.12. Дана 2-КНФ формула на $n + m$ переменных. Проверьте, правда ли, что для любых значений первых n переменных найдется такой набор значений последних m переменных, что формула будет принимать значение 1.

2.13. Придумайте **полиномиальный** алгоритм поиска лексикографически минимального решения `2-SAT` (поиска лексикографически минимального набора значений переменных, удовлетворяющего формулу).

2.14. Найдите правильную раскраску неориентированного графа в три цвета, если для каждой вершины дан один цвет, в который ее запрещено красить.

2.15. Есть n городов, расположенных на окружности по часовой стрелке в порядке возрастания номеров. Каждые два соседних города (i и $i+1$ или 1 и n) соединены дорогой. Требуется построить еще m дорог между определенными городами, при чем каждую из них можно провести либо внутри окружности, либо снаружи. Определите, можно ли построить все необходимые дороги, чтобы никакие две не пересекались, за время $\mathcal{O}(n^2 + m)$.

2.16. **Радиостанции.** Есть n радиостанций, изначально все выключены. Есть m заявок вида «включите радиостанцию a_i или b_i». Есть k ограничений «одна из радиостанций c_i и d_i» должна оставаться выключенной. Кроме того, у каждой радиостанции есть два параметра l_i и r_i $(0 \leqslant l_i \leqslant r_i \leqslant F)$.

Требуется включить некоторые радиостанции и выбрать общую мощность f, чтобы выполнялись все описанные требования, а для каждой включенной радиостанции выполнялось $l_i \leqslant f \leqslant r_i$.

Решите задачу за время

(a) $\mathcal{O}(n^2 + m + k)$
(b) $\mathcal{O}((n + m + k) \cdot F)$
(c) $\mathcal{O}(n + m + k + F)$
(d) $\mathcal{O}(n \log n + m + k)$

Неделя 3. Мосты, ТС, компоненты двусвязности

Практика

3.1. Строго докажите, почему при поиске мостов или точек сочленения можно использовать как $\texttt{depth}[v]$, так и $\texttt{time}_{in}[v]$ с одинаковой эффективностью.

3.2. Правда ли, что

(a) все ребра, выходящие из точки сочленения – мосты?

(b) из каждой точки сочленения выходит хотя бы один мост?

(c) если все выходящие из вершины ребра – мосты, то она – точка сочленения?

(d) если из вершины выходят хотя бы два моста, то она – точка сочленения?

(e) если вершина лежит на простом цикле, то она не может быть точкой сочленения?

3.3. Дан неориентированный граф. За время $\mathcal{O}(n+m)$ найдите количество ребер, которое необходимо в него добавить, чтобы он стал реберно двусвязным.

3.4. Какое минимальное и максимальное число

(a) мостов

(b) точек сочленения

может быть в графе из n вершин?

3.5. В неориентированном графе на каждом ребре написано число. Научитесь отвечать на запросы вида «какое минимальное число можно встретить, идя по пути из u в v» за $\mathcal{O}(n \log n + m)$ предподсчета и $\mathcal{O}(\log n)$ на запрос. Пути рассматриваем

(a) реберно простые

(b) вершинно простые

3.6. В данном связном неориентированном графе найдите количество способов удалить два ребра так, чтобы граф стал несвязным, за время

(a) $\mathcal{O}(n^2)$

(b) $\mathcal{O}(m)$

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

Эйлеровость

3.8. Добавьте в

(a) связный неориентированный

(b) связный ориентированный

(c) несвязный неориентированный

(d) несвязный ориентированный

граф минимальное число ребер так, чтобы граф стал эйлеровым (стал содержать эйлеров цикл).

3.9. Покажите, как за время $\mathcal{O}(n+m)$ разбить все ребра графа на минимальное количество реберно простых путей.

Подсказка: сначала дополните граф до эйлерового.

3.10. Вам дан набор из n слов. Проверьте, можно ли упорядочить эти слова так, чтобы каждое следующее начиналось на ту же букву, на которую заканчивается предыдущее, за время $\mathcal{O}(n)$.

3.11. За время $\mathcal{O}(2^n \cdot n)$ постройте битовую строку минимальной длины, в которой в качестве подстрок содержатся все возможные битовые строки длины n.

Например, «`01100`» содержит в качестве подстрок «`00`», «`01`», «`10`» и «`11`».

3.12. Есть комнаты и двери между комнатами. Нужно 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. Решить за время $\mathcal{O}(n + m)$.

3.13. `[Интернет-олимпиады 2021]`. Дан неориентированный граф. Ориентируйте его ребра так, чтобы величина $\sum\limits_{v=1}^{n} \mathtt{deg}_{in}(v) \cdot \mathtt{deg}_{out}(v)$ была максимальна. Время работы $\mathcal{O}(n+m)$.

Больше всяких задач на DFS

3.14. На каждом ребре ориентированного графа написана буква (не обязательно из каждой вершины выходят ребра с различными буквами). За полиномиальное от размера графа и длины строки время найдите все вершины, из которых существует путь, собирающий данную строку s.

3.15. Дан ориентированный граф, в каждой вершине находятся либо монстры одного из трех типов, либо иммунитет к монстрам одно из трех типов. Иммунитет одноразовый и пропадает после прохождения через вершину с монстрами. За время $\mathcal{O}(n + m)$ определите, можно ли добраться из вершины a в вершину b и не умереть.

3.16. Придумайте способ хранения графа, занимающий $\mathcal{O}(m)$ (независимо от n) памяти, и DFS на котором работает за $\mathcal{O}((n + m) \cdot \mathrm{poly}(\log n))$.

Неделя 4. Неделя «чилла»

Практика

4.1. Дан DAG на n вершинах и m ребрах.

(a) Постройте лексикографически минимальную топологическую сортировку за время $\mathcal{O}(n+m)$.

(b) Посчитайте количество топологических сортировок в графе $\mathcal{O}(\text{poly}(n+m))$.

4.2. Дан **ориентированный** граф на n вершинах и m ребрах.

(a) Найдите в нем цикл, проходящий по конкретному ребру, за время $\mathcal{O}(n+m)$.

(b) Найдите в нем цикл, проходящий через конкретную вершину, за время $\mathcal{O}(n+m)$.

4.3. Дан **неориентированный** граф на n вершинах и m ребрах.

(a) Найдите в нем цикл, проходящий по конкретному ребру, за время $\mathcal{O}(n+m)$.

(b) Найдите в нем цикл, проходящий через конкретную вершину, за время $\mathcal{O}(n+m)$.

4.4. За время $\mathcal{O}(n+m)$ определите в дереве

(a) количество путей, проходящих через каждое ребро;

(b) суммарную длину путей, проходящих через каждое ребро.

4.5. Дан

(a) DAG;

(b) произвольный ориентированный граф.

За время $\mathcal{O}(n+m)$ определите, существует ли путь из четного числа ребер из вершины v в вершину u.

4.6. Подвешенное дерево на n вершинах хранится как массив родителей каждой вершины. Как поменяется этот массив, если переподвесить это дерево за другую вершину v (сделать ее корнем)?

> **Def.** *Центроидом* дерева на n вершинах называется такая вершина v, при удалении которой из дерева все компоненты связности будут размера $\leqslant \frac{n}{2}$.

4.7. Докажите эквивалентность данного определения центроида следующему определению: «v – центроид дерева, если величина $\sum\limits_{u=1}^{n} \texttt{dist}(u, v)$ минимальна по всем v».

4.8. Докажите, что

(a) в дереве бывает не более двух центроидов;

(b) если их два, то они – смежные (связаны ребром).

4.9. Придумайте алгоритм поиска центроида дерева, работающий за время $\mathcal{O}(n+m)$.

4.10. Дан неориентированный граф на n вершинах и m ребрах. За время $\mathcal{O}(m^2)$ ориентируйте его ребра так, чтобы максимум исходящих степеней вершин был как можно меньше.

Подсказка: Ориентируйте ребра как-нибудь, затем попробуйте сколько-то раз улучшить результат, поменяв ориентацию ребер вдоль определенных путей.

Def. *Диаметром* дерева называется его самый длинный путь.

4.11. Придумайте алгоритм, который находит диаметр дерева за время $\mathcal{O}(n+m)$.

4.12. **Гроб гроб кладбище**. Дан неориентированный граф. Разбейте все его вершины за время $\mathcal{O}(n+m)$ на три попарно непересекающихся множества L, Mи R так, чтобы

- ▷ M было путем в графе;
- ▷ между вершинами L и R не было путей, не проходящих через хотя бы одну вершину M;
- ▷ выполнялось $|L| = |R|$.

Домашнее задание

★ **Правила сдачи**

- ▷ Письменное ДЗ оформляется в LaTeX, написанные от руки решения приниматься не будут
- ▷ Решения надо прислать **до 8 октября 2023, 23:59** на почту `itmo.algo.teaching+y2021@gmail.com`
- ▷ Тема письма должна быть указана в виде «`Группа HW01 Фамилия Имя`» (без кавычек), например: «`M3230 HW01 Иванов Иван`». Обратите внимание, что буква «M» в названии группы – латинская
- ▷ К письму должны быть приложены как сгенерированный `.pdf`-файл, так и исходный `.tex`-файл

1. Расскажите смешной не-баянистый анекдот.

2. Дана матрица чисел размера $n \times m$. За время

   (a) $\mathcal{O}(n^2m^2)$
   (b) $\mathcal{O}(nm)$

   найдите наибольшее связное по сторонам множество клеток, в которых встречается не более двух различных чисел.

3. Во взвешенном (у каждого ребра есть вес) графе проведены «синие» и «красные» ребра. Человек изначально может перемещаться только по синим ребрам. Чтобы «переключиться» на возможность перемещаться только по красным ребрам, требуется x времени. И наоборот, если сейчас ему доступны только красные ребра, он может потратить x времени, и получить возможность перемещаться только по синим. За какое минимальное время можно добраться из u в v? Время решения – $\mathcal{O}(n+m)$.

4. Есть n гирек и результаты их m взвешиваний. Каждое взвешивание дает знак (‘<’, ‘=’, ‘>’) между весами каких-то двух гирек. Определите, не противоречивы ли результаты, и, если противоречивы, после какого взвешивания впервые возникает противоречие. Время работы $\mathcal{O}((n+m)\log m)$.

5. Дан ориентированный граф, исходящая степень каждой вершины равна 1. Научитесь в таком графе

   (a) за $\mathcal{O}(n)$ находить для каждой вершины самый длинный простой путь, начинающийся в ней;
   (b) за $\mathcal{O}(1)$ отвечать на запросы вида «куда мы попадем, если пройдем k шагов вперед из вершины v?».

Неделя 5. Кратчайшие пути

Практика

BFS

5.1. Рассмотрим алгоритм поиска кратчайшего пути в графе с отрицательными весами: если все веса лежат в интервале $[-a, b]$, сначала увеличим вес каждого ребра на a, а затем запустим 0-$(a+b)$-`bfs`. Приведите контрпример к этому алгоритму.

5.2. Дан невзвешенный ориентированный граф. За время $\mathcal{O}(n+m)$ определите для каждого ребра графа, проходит ли через него хотя бы один кратчайший путь из s в t.

5.3. Дан невзвешенный ориентированный граф. За время $\mathcal{O}(n+m)$ найдите количество кратчайших путей из s в t.

5.4. Дан невзвешенный ориентированный граф. За время $\mathcal{O}(n+m)$ найдите минимальное количество ребер, которые надо развернуть, чтобы в графе появился путь из s в t.

5.5. Дан невзвешенный ориентированный граф. За время $\mathcal{O}(n+m)$ найдите вершину, суммарное расстояние от которой до s и до t минимально.

5.6. Дан невзвешенный ориентированный граф. За время $\mathcal{O}(n+m)$ найдите, какое ребро надо развернуть, чтобы в итоговом графе кратчайший путь из s в t был как можно меньше.

5.7. Калькулятор умеет делать две операции: $x \to (x+3) \bmod M$ и $x \to 4x \bmod M$. За какое минимальное число операций можно получить из числа a число b? Время работы $\mathcal{O}(M)$.

5.8. Дан невзвешенный неориентированный граф. Найдите кратчайший путь из какой-либо вершины из множества вершин A в какую-либо вершину множества B за время $\mathcal{O}(n+m)$.

5.9. Дана шахматная доска 30×30 и позиции двух коней на ней. Определите, за какое минимальное число ходов можно поменять этих двух коней местами.

Кратчайшие пути на взвешенных графах

5.10. Предложите способ модифицировать алгоритм Дейкстры, если весом пути считать

(a) произведение весов ребер на пути

(b) максимальное из ребер на пути

Какие ограничения на веса ребер следует установить, чтобы получившийся алгоритм работал корректно?

5.11. Рассмотрим кучу, в которой `add` и `extractMin` все так же работают за $\mathcal{O}(\log n)$, но `decreaseKey` работает за $\mathcal{O}(1)$. За какое время будет работать алгоритм Дейкстры, реализованный на такой куче? Какое d следует выбирать, чтобы алгоритм Дейкстры на d-ичной куче работал оптимально?

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

5.13. Модифицируйте произвольный алгоритм поиска кратчайшего пути, не меняя асимптотику времени решения и используемой памяти, чтобы он из всех кратчайших путей находил

(a) путь из минимального числа ребер

(b) путь с наименьшим весом максимального ребра на нем

5.14. Пусть вес пути — это минимальный из весов ребер на нем. Предложите алгоритм поиска кратчайших путей от вершины s до всех остальных в этом графе, работающий за время $\mathcal{O}(m\log(\text{answer}))$.

5.15. Дан взвешенный граф с положительными весами ребер. Найдите кратчайшие пути от вершины s до всех остальных, если на каждую вершину есть дополнительное ограничение: через вершину v можно проходить только в периоды длиной a_v, между которыми она «закрыта» на b_v времени. Время работы — $\mathcal{O}(m\log n)$ или $\mathcal{O}(n^2)$.

5.16. Дан взвешенный граф, в котором за единицу времени можно пройти сколько угодно ребер, если их суммарный вес не превышает d. Найдите в нем кратчайшие пути от стартовой вершины s до остальных за время $\mathcal{O}(m\log n)$.

5.17. Определите максимальный суммарный вес ребер графа, которые можно выкинуть, чтобы вершины s и t остались связаны путем, за время $\mathcal{O}(m\log n)$.

5.18. Приведите пример графа без отрицательных циклов, на котором алгоритм Флойда с порядком циклов `i`, `j`, `k` работает неверно.

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

5.20. Модифицируйте алгоритм Дейкстры, чтобы он работал за время $\mathcal{O}(m+d)$, если известно, что все кратчайшие пути имеют длину не больше d.

5.21. Модифицируйте алгоритм Форда-Беллмана, чтобы он находил кратчайшие пути, состоящие из от l до r ребер.

5.22. Дана матрица кратчайших расстояний между каждой парой вершин. Постройте граф с неотрицательными весами ребер, которому соответствует эта матрица, или определите, что такой граф не существует. Время работы $\mathcal{O}(\text{poly}(n))$.

5.23. Дана матрица кратчайших расстояний между каждой парой вершин. Требуется за время $\mathcal{O}(n^2)$ пересчитывать ее при запросах «вес ребра между u_i и v_i уменьшился на x_i».

5.24. Докажите, что задача поиска кратчайшего **простого** пути в графе с отрицательными циклами является `NP`-полной.

Подсказка: сведите задачу поиска гамильтонова пути к данной.

Неделя 6. Минимальные остовы

Практика

6.1. Докажите, что минимальное по весу остовное дерево также имеет минимально возможное максимальное ребро.

6.2. Рассмотрим в связном неориентированном графе путь из v в u, в котором вес максимального ребра наименьший возможный. Докажите или опровергните, что существует минимальное остовное дерево,

(a) содержащее все ребра этого пути

(b) максимальное ребро на пути между u и v в котором не превосходит максимальное ребро на рассмотренном пути

6.3. В алгоритме Дейкстры в качестве метрики, по которой выбирается следующая вершина, используется $\texttt{d}[v] = \texttt{shortest\_path}(s, v)$. В алгоритме Прима используется $\texttt{d}[v] = \texttt{min\_edge}(\to v)$.
Докажите или опровергните, что если использовать в качестве метрики «вес ребра из последней обработанной вершины в v», то такой алгоритм будет в полном взвешенном графе находить минимальный гамильтонов путь, начинающийся в выбранной стартовой вершине.

6.4. Даны n точек на плоскости. Требуется найти минимальное такое d, чтобы, прыгая между этими точками на расстояние не больше d, можно было добраться из точки номер 1 в точку номер n. Решите за время

(a) $\mathcal{O}(n^2 \log(\text{answer}))$ без использования алгоритмов поиска MST

(b) $\mathcal{O}(n^2)$

6.5. Вам нужно передать с одного компьютера на другой n файлов. Каждый файл – это битовый вектор размера m. Файл можно переслать либо просто как последовательность бит, либо как `diff` с другим файлом, уже пересланным ранее. В первом случае нужно будет передать m бит, во втором – $A + B \cdot d$ бит, где d – число различающихся бит в файлах, A и B – некоторые константы. Найдите минимальное число бит, которое нужно передать. Порядок передачи файлов не фиксирован. Время работы $\mathcal{O}(n^2)$.

6.6. Даны n точек на плоскости, соответствующих вершинами графа. Между ними провели $n - 1$ ребро так, что образовалось ровно k компонент связности. Требуется за время $\mathcal{O}(n^2)$ определить, какие $k-1$ замен ребер следует сделать (удалить и провести любое другое ребро), чтобы получить минимальное возможное остовное дерево.

6.7. Есть n городов, изображенных точками на плоскости. Можно соединить два города дорогой, потратив $A \cdot \text{len}$ денег, где len – длина дороги. Также можно построить в городе аэропорт, потратив B денег. За время $\mathcal{O}(n^2)$ определите минимальное число денег, которые надо потратить, чтобы от любого города можно было добраться до любого другого.

6.8. Для данного неориентированного графа определите за время $m \log m + \mathcal{O}(m\alpha(n))$ для каждого ребра

(a) существует ли минимальный остов, содержащий его

(b) существует ли минимальный остов, не содержащий его

6.9. Найдите количество минимальных остовов в данном неориентированном графе за время $\mathcal{O}(\text{poly}(m))$.

6.10. За время $\mathcal{O}(m \log n)$ найдите минимальное остовное дерево, в котором степень вершины 1 не превосходит d.

6.11. Докажите, что задача о поиске минимального остовного дерева с ограничением, что для каждой вершины v ее степень в этом остове не превышает $\mathtt{d}[v]$, не проще задачи поиска минимального гамильтонова пути/цикла.

6.12. Приведите пример графа из n вершин и m ребер, на котором алгоритм Борувки работает за

(a) $\Omega(m \log n)$

(b) $\mathcal{O}(n)$

6.13. Докажите, что алгоритм Борувки на планарных графах работает за $\mathcal{O}(n)$.

> **Def.** Назовем *ориентированным остовным деревом* (*Arborescence*) из вершины v такое множество из $n-1$ ребра в ориентированном графе, что по его ребрам можно попасть из v в любую другую вершину.

6.14. Покажите, что аналог алгоритма Прима (каждый раз выбирать минимальное ребро, ведущее из уже построенного дерева в оставшиеся вершины), не работает для построения минимального ориентированного остовного дерева.

Математические мемы

6.15. **Формула Вилланса**. Докажите следующую формулу для вычисления n-го простого числа:

$$p_n = 1 + \sum_{i=1}^{2^n} \left\lfloor \left(\frac{n}{\sum\limits_{j=1}^{i} \left\lfloor \left(\cos \pi \frac{(j-1)!+1}{j} \right)^2 \right\rfloor} \right)^{\frac{1}{n}} \right\rfloor$$

(a) **Теорема Вильсона**. Докажите, что $(p-1)!+1$ делится на p тогда и только тогда, когда $p \in \mathbb{P}$.

(b) Опишите поведение выражения $\left\lfloor \left(\cos \pi \frac{(j-1)!+1}{j} \right)^2 \right\rfloor$ в зависимости от простоты j.

(c) Сформулируйте, какую численную характеристику числа i вычисляет выражение в знаменателе формулы Вилланса.

(d) Пользуясь предыдущими пунктами, докажите формулу Вилланса.
Подсказка: 2^n в качестве верхней границы суммирования берется, основываясь на постулате Бертрана.

Неделя 7. Игры на графах

Практика

7.1. Приведите пример игры на графе, в которой

(a) существует более одного сценария игры, но вне зависимости от ходов первого игрока второй всегда проигрывает

(b) количество возможных сценариев игры экспоненциально от числа вершин графа, при чем первый ход однозначно определяет победителя вне зависимости от следующих ходов обоих игроков

7.2. Дан прямоугольный параллелепипед размера $a \times b \times c$. За один ход игрок должен сделать разрез любой из имеющихся его частей вдоль любого направления, паралельного осям координат, по целой координате. Проигрывает не имеющий хода. Кто из k игроков выигрывает в такой игре?

7.3. Предыдущая игра, но теперь игроков двое, и после разреза игрок должен выкинуть из игры одну из двух полученных частей на свой выбор. Как определить, кто выигрывает в такой игре, за время $\mathcal{O}(\max(a, b, c)^2)$?

7.4. Придумайте интересную «нечестную» игру, в которой при фиксированных параметрах вне зависимости от ходов всегда побеждает конкретный игрок. Чем менее очевидна нечестность вашей игры, тем больше баллов вы получите за задание. Эту задачу надо сдавать в телеграм @doreshnikov, но каждый следующий сдающий должен сдавать что-то новое.

7.5. Найдите победителя в игре на графе, если в графе могут быть циклы, но наложено ограничение, что нельзя перемещать фишку

(a) в вершину, в которой она уже находилась

(b) по ребру, по которому она уже была перемещена

7.6. Дана игра на графе из n вершин и m ребер, в которой ход заключается в перемещении фишки на

(a) два ребра вперед

(b) одно или два ребра вперед на выбор игрока

Определите, кто из двух игроков выигрывает при правильной игре, за время $\mathcal{O}(n^2)$.

7.7. Дано две игры на графах, но вместо их суммы будем рассматривать игру, в которой

(a) игрок делает ход в обеих играх одновременно

(b) игроки делают ходы в первой игре, но как только в ней ход становится невозможным, тот игрок, чей сейчас ход, начинает своим ходом вторую игру

Определите, кто из двух игроков выигрывает при правильной игре, за время $\mathcal{O}(n+m)$, где n и m – суммарное число вершин и ребер в играх.

7.8. Рассмотрим произвольную игру на графе. Первый игрок хочет, чтобы игра закончилась (фишка попала в тупик), а второй игрок хочет, чтобы игра никогда не заканчивалась. Определите за время $\mathcal{O}(n + m)$, кто из них сможет добиться своего.

7.9. Рассмотрим игру на графе, в которой каждый игрок считает ничью самым плохим результатом, поэтому, если у него нет возможности выиграть, он будет стремиться проиграть вместо того, чтобы затягивать игру перемещениями по циклу. Что не так в такой игре?

7.10. Почему нельзя анализировать асимметричные игры на графе (с ребрами двух цветов) с помощью функции Гранди в явном виде, не сводя их к симметричным с удвоенным количеством состояний?

7.11. Рассмотрим игру в «поддавки» – стандартная игра на графе, но выигрывает тот, кто не может ходить.

(a) Как свести такую игру к игре со стандартными правилами: «проигрывает не имеющий хода»?

(b) Можно ли анализировать сумму таких игр с помощью функции Гранди, предварительно сведя каждую из них к игре со стандартными правилами?

7.12. Рассмотрим игру A, для всех состояний которой была посчитана функция Гранди. Опишите выигрышную стратегию второго игрока в игре $A + \texttt{Nim}(\mathrm{grundy}(A))$.

7.13. Дана прямоугольная полоса размера $1 \times n$. За ход можно поставить крестик в любую ее клетку, которая еще не занята крестиком, и оба соседа которой тоже не заняты крестиком. Проигрывает не имеющий хода. За время $\mathcal{O}(n^2)$ определите, кто из двух игроков выигрывает при правильной игре.

7.14. Та же задача, но теперь крестик можно ставить в любую незанятую ячейку, а выигрывает тот, после чьего хода на полоске есть три крестика подряд. Кто выигрывает в такой игре?

7.15. Дано клетчатое поле, некоторые клетки заблокированы, в одной из свободных клеток стоит фишка. За ход можно переместить фишку в свободную клетку, соседнюю с текущей по стороне, но нельзя перемещать фишку в клетку, в которой она уже была. Кто выигрывает при правильной игре?

7.16. Дана изначально пустая последовательность и массив целых чисел длины n. За ход игрок может стереть число из начала или конца массива и записать его в конец последовательности. Проигрывает тот, после чьего хода последовательность перестает быть возрастающей. За время $\mathcal{O}(n)$ определите, кто выигрывает при правильной игре.

Неделя 8. Строки. Хеши, префикс-функция, $\mathfrak{z}$-функция

Практика

8.1. Оцените вероятность коллизии хешей хотя бы двух подстрок, если хеши равномерно распределены по $[0, M)$ и совершается q запросов на подстроках строки длины n.

> **Def.** Число k называется *бордером* строки s, если ее префикс и суффикс длины k равны. Также под бордером можно понимать и сам соответствующий префикс или суффикс длины k.

> **Def.** Число k называется *периодом* строки s, если $s_i = s_{i+k}$ для всех i. Также под периодом можно понимать и соответствующий префикс строки длины k.

8.2. Докажите, что если k – бордер строки длины n, то $n - k$ – ее период.

8.3. Найдите все периоды строки длины n за время $\mathcal{O}(n)$, используя

(a) хеши

(b) префикс-функцию

(c) $\mathfrak{z}$-функцию

8.4. **Вхождение со сдвигом**. Даны два массива s и t длины $\leqslant n$. Требуется найти все вхождения массива вида $[t_0+\delta, t_1+\delta, t_2+\delta, \ldots]$ в s, где δ может быть любой, за время $\mathcal{O}(n)$.

(a) Элементы массивов от 0 до 26. Можно считать, что деление по модулю происходит за $\mathcal{O}(1)$.

(b) Элементы произвольные.

> Здесь и далее будем использовать обозначения $s^{i\to}$ для суффикса строки s, начинающегося с символа номер i, и $s_{\to i}$ для префикса строки s, заканчивающегося **строго до** символа i. Стандартная нумерация символов от 0 до $|s| - 1$.

> **Def.** $\texttt{lcp}(s, t)$ (*longest common prefix*) – длина наибольшего общего префикса строк s и t.

8.5. Дана строка s длины n. Посчитайте матрицу значений $\texttt{lcp}(s^{i\to}, s^{j\to})$ по всем i и j за время $\mathcal{O}(n^2)$.

8.6. С помощью матрицы из предыдущего задания найдите количество различных подстрок на каждой подстроке строки s за время $\mathcal{O}(n^2)$.

8.7. Дана строка s длины n. Требуется отвечать на запросы

(a) является ли заданная подстрока палиндромом, за время $\mathcal{O}(1)$

(b) является ли заданная подстрока палиндромом, за время $\mathcal{O}(\log n)$ без использования хэшей

(c) являются ли две заданные строки анаграммами, за время $\mathcal{O}(|\Sigma|)$

(d) являются ли две заданные строки анаграммами, за время $\mathcal{O}(1)$

и $\mathcal{O}(n)$ на предподсчет ($\mathcal{O}(n|\Sigma|)$ в пункте (e)).

8.8. Даны две строки s и t длины не больше n. Найдите их наибольшую (по длине) общую подстроку за время $\mathcal{O}(n \log n)$.

8.9. Дана строка s длины n. Требуется обрабатывать запросы

- ▷ изменить символ строки
- ▷ проверить равенство двух подстрок

за время $\mathcal{O}(\log n)$ на запрос и $\mathcal{O}(n \log n)$ на предподсчет.

8.10. Дана строка s длины n. Требуется обрабатывать запросы

- ▷ вырезать подстроку и вставить ее в другое место строки
- ▷ проверить равенство двух подстрок

за время $\mathcal{O}(\log n)$ на запрос и $\mathcal{O}(n \log n)$ на предподсчет.

8.11. `IO 09.10.2022 H`. Дана строка s длины n. Требуется обрабатывать запросы

- ▷ изменить символ строки, за время $\mathcal{O}(\log n)$
- ▷ заменить все вхождения символа c_1 на символ c_2, за время $\mathcal{O}(\log^2 n)$ амортизированно
- ▷ проверить равенство двух подстрок, за время $\mathcal{O}(|\Sigma|)$

и $\mathcal{O}(n \log n)$ на предподсчет.

> **Def.** *Тандемным повтором* называется любая строка, являющаяся конкатенацией двух одинаковых строк.

8.12. За время $\mathcal{O}(n \log n)$ найдите в строке длины n все подстроки, являющиеся тандемными повторами,

(a) используя хеши
(b) не используя хеши

8.13. Для строк s и t длины n и перестановки чисел от 1 до n определите за время $\mathcal{O}(n)$, сколько минимум раз надо применить перестановку к символам строки s, чтобы получить строку t.

> В следующих заданиях запрещается использовать хеши.

8.14. Дана строка s длины n. Требуется для каждого ее префикса

(a) проверить, является ли он палиндромом
(b) найти количество его вхождений в s
(c) найти количество его бордеров

за время $\mathcal{O}(n)$.

8.15. За время $\mathcal{O}(n)$ постройте

(a) строку с заданной префикс-функцией
(b) строку с заданной $\mathcal{Z}$-функцией
(c) $\mathcal{Z}$-функцию строки по заданной префикс-функции, не восстанавливая саму строку
(d) префикс-функцию по заданной $\mathcal{Z}$-функции, не восстанавливая саму строку

8.16. Для данной строки s длины n найдите число ее суффиксов, которые лексикографически меньше ее, за время $\mathcal{O}(n)$.

8.17. **Вхождение с заменой**. Назовем две строки *эквивалентными*, если одну можно получить из другой заменой, при которой не теряется отношение равенства и неравенства символов, например «`abacaba`» и «`xyzqxyz`». Найдите все вхождения в строку s строк, эквивалентных заданной t, за время $\mathcal{O}(|s| + |t|)$.

8.18. **Сжатие строки**. Будем обозначать строкой `x(s)` повторенную x раз строку s. Например, «`2(a3(bc)d)`» соответствует строке «`abcbcbcdabcbcbcd`». Для данной строки длины n найдите самую короткую ее запись в таком виде за время $\mathcal{O}(n^3)$.

> **Def.** *Строка Туэ-Морса* – бесконечная строка, получаемая как $\lim\limits_{n\to+\infty} t_n$, где t_n определяется рекуррентно:
>
> ▷ $t_0 =$ «`0`»
>
> ▷ $t_{i+1} = t_i + \overline{t_i}$, где $\overline{t_i}$ – замена в t_i всех 0 на 1 и наоборот

8.19. Докажите, что i-й символ строки Туэ-Морса соответствует четности числа единиц в двоичной записи числа i.

8.20. Докажите, что никакие две равные подстроки строки Туэ-Морса не могут иметь пересекающиеся вхождения в нее.

8.21. Модифицируйте алгоритм нахождения $\aleph$-функции для нахождения массивов p^0 и p^1, где p_i^0 – длина максимальной подстроки-палиндрома с центром в i-м символе строки, а p_i^1 – длина максимальной подстроки-палиндрома с центром между i-м и $i+1$-м символами строки

8.22. Докажите, что в строке длины n может быть не более n различных подстрок-палиндромов.

Неделя 9. Строковые автоматы. Бор и автомат Ахо-Корасик

Практика

Бор

> Во всех следующих задачах, если не указано обратное, считайте размер алфавита константным.

9.1. Дан набор строк $s_1, \ldots, s_n$. Отсортируйте эти строки в лексикографическом порядке за время $\mathcal{O}(\sum |s_i|)$.

9.2. Дан набор строк $s_1, \ldots, s_n$. Требуется найти самую короткую строку, не являющуюся префиксом никакой из них, за время $\mathcal{O}(\sum |s_i|)$.

9.3. Дан набор строк $s_1, \ldots, s_n$. Обозначим за L суммарную длину этих строк. Докажите, что

(a) для любой строки t множество значений $\texttt{lcp}(t, s_i)$ имеет размер $\mathcal{O}(\sqrt{L})$

(b) у любой вершины в боре на строках s_i есть $\mathcal{O}(\sqrt{L})$ терминальных вершин-предков

(c) у любой вершины в боре на строках s_i есть $\mathcal{O}(\sqrt{L})$ вершин-предков, у которых хотя бы два ребенка

9.4. Дан набор строк $s_1, \ldots, s_n$ суммарной длины S. Найдите количество пар индексов (i, j) таких, что $s_i + s_j$ – палиндром, за время $\mathcal{O}(S)$,

(a) используя хеши

(b) не используя хеши

9.5. Дан набор неотрицательных целых чисел $a_1, \ldots, a_n$. Все числа не превосходят C. Используя бор, научитесь отвечать на следующие запросы за $\mathcal{O}(\log C)$:

(a) добавить число в набор/удалить число из набора

(b) найти k-ю порядковую статистику

(c) для данного числа x найти такое i, что $x \texttt{ xor } a_i$ максимально

(d) найти $\texttt{xor}$ всех чисел из набора, меньших данного x

9.6. Приведите пример множества строк, для которого сжатый бор имеет больше вершин, чем минимальный автомат, допускающий только строки этого множества.

9.7. Постройте детерминированный автомат, допускающий все десятичные записи чисел, делящихся на k, имеющий

(a) $\mathcal{O}(k)$ состояний, при чем числа передаются в автомат слева-направо

(b) $\mathcal{O}(k^2)$ состояний, при чем числа передаются в автомат справа-налево

9.8. Дана строка s длины n. Постройте

(a) детерминированный автомат, допускающий все префиксы строки s, имеющий $\mathcal{O}(n)$ состояний

(b) недетерминированный автомат, допускающий все суффиксы строки s, имеющий $\mathcal{O}(n)$ состояний

9.9. Дан набор строк $s_1, \ldots, s_n$. Постройте детерминированный автомат, допускающий все строки, кроме строк данного набора, имеющий $\mathcal{O}(\sum |s_i|)$ состояний.

Автомат Ахо-Корасик

9.10. Дан набор слов $s_1, \ldots, s_n$. Научитесь за время $\mathcal{O}(|t| + \sum |s_i|)$ для данного текста t находить количество вхождений, первое и последнее вхождение в него каждого из слов.

9.11. Дан набор слов $s_1, \ldots, s_n$ суммарной длины S.

(a) Посчитайте количество строк длины m, содержащих хотя бы одно из этих слов в качестве подстроки за время $\mathcal{O}(mS)$.

(b) Постройте автомат размера $\mathcal{O}(S)$, допускающий только строки, содержащие хотя бы одно из этих слов в качестве подстроки.

(c) Предложите алгоритм, определяющий по данной строке t, входят ли в нее все s_i в качестве подстрок, за время $\mathcal{O}(|t| + S)$.

(d) Постройте автомат размера $\mathcal{O}(2^n \cdot S)$, допускающий только строки, содержащие все данные слова в качестве подстрок.

9.12. Дан построенный автомат Ахо-Корасик со всеми переходами и суффиксными ссылками, на котором стерты все символы переходов. За полиномиальное от его размера время восстановите любое назначение символов переходам, для которых автомат является корректным.

9.13. Построен сжатый бор на наборе строк $s_1, \ldots, s_n$ суммарной длины S. Предложите способ определить суффиксные ссылки для его вершин. Постройте их за время

(a) $\mathcal{O}(S)$

(b) $\mathcal{O}(n)$

9.14. Даны строки t, s_1 и s_2 длины не больше n. За время $\mathcal{O}(n)$ проверьте, существуют ли

(a) непересекающиеся

(b) непересекающиеся, но для трех строк s_1, s_2 и s_3

(c) пересекающиеся

вхождения s_i в t.

9.15. Дана строка t и набор строк $s_1, \ldots, s_n$ суммарной длины S.

(a) За время $\mathcal{O}(|t| + S)$ найдите наибольшее количество попарно непересекающихся вхождений s_i в t.

(b) за время $\mathcal{O}(|t|^2 + S)$ проверьте, существует ли разбиение t на подстроки из набора.

9.16. Будем говорить, что строка t покрывается набором строк s_i, если каждый символ t содержится в некотором вхождении какого-то из s_i в t. Дана строка t и набор строк суммарной длины S. За время $\mathcal{O}(\mathrm{poly}(|t| + S))$ проверьте, покрывается ли данная строка данным набором.

9.17. За линейное от размера ввода время найдите все вхождения прямоугольниной подматрицы в другую матрицу.

9.18. Дан набор строк $s_1, \ldots, s_n$. Строки отсортированы лексикографически, для каждой пары соседних строк известен их $\mathtt{lcp}(s_i, s_{i+1})$. Постройте сжатый бор для этих строк за время $\mathcal{O}(n)$.

9.19. **Суффиксные кросс-ссылки**. Дано два бора T_1 и T_2 размера не более n. Для каждой вершины $u \in T_1$ найдите самую глубокую вершину $v \in T_2$, что $\mathtt{str}(v)$ – суффикс $\mathtt{str}(u)$, за время $\mathcal{O}(n)$.

9.20. Даны n шаблонов, каждый символ шаблона – либо обычный символ, либо wildcard на один (‘`.`’) или произвольное число (‘`*`’) символов. Существует ли бесконечная (вправо) строка, не содержащая ни одного шаблона в качестве подстроки? Определите за полиномиальное от размера ввода время.

Неделя 10. Суффиксный массив

Практика

10.1. Докажите, что число различных подстрок в строке равно

$$\frac{n(n+1)}{2} - \sum_{i=1}^{n} \texttt{lcp}[i]$$

10.2. За время $\mathcal{O}(n \log n)$ посчитайте сумму длин всех различных подстрок строки длины n.

10.3. Дана строка s длины n. Требуется отвечать на запросы «найти число вхождений подстроки t в s» за $\mathcal{O}(|t| + \log n)$.

10.4. Дана строка s длины n. Строка S определяется как s^k (k повторений) для некоторого целого k. Требуется отвечать на запросы «найти число вхождений подстроки t в S» за $\mathcal{O}(|t| + n \log n)$.

10.5. Дана строка s длины n и ее суффиксный массив. Покажите как за время $\mathcal{O}(n)$ найти самую длинную подстроку, имеющую

(a) хотя бы k вхождений в s

(b) хотя бы два непересекающихся вхождения в s

10.6. **Рефрен**. Дана строка s длины n. За время $\mathcal{O}(n \log n)$ найдите ее подстроку, для которой максимально произведение ее длины и количества ее вхождений в s.

10.7. Дан набор строк s_i суммарной длины n. Найдите их наибольшую общую подстроку за время $\mathcal{O}(n \log n)$ (не используя хеши).

10.8. Дан набор строк s_i суммарной длины n. Разрешается каждую строку циклически сдвинуть. Требуется максимизировать длину наибольшего общего префикса s_i за время $\mathcal{O}(n)$, если есть blackbox, который умеет за $\mathcal{O}(n)$ строить суффиксный массив.

10.9. Дана строка s длины n, для которой уже построен суффиксный массив и `lcp`. Найдите

(a) количество различных подстрок на каждом суффиксе s за время $\mathcal{O}(n \log n)$

(b) количество различных подстрок на каждом суффиксе s за время $\mathcal{O}(n)$

(c) количество различных подстрок на каждом префиксе s за время $\mathcal{O}(n \log n)$

(d) количество различных подстрок на каждом префиксе s за время $\mathcal{O}(n)$

10.10. К строке s не был дописан в конец специальный символ. После алгоритмом построения суффиксного массива сортируются циклические сдвиги исходной строки. Приведите пример s, на которой на полученном массиве алгоритм Касаи найдет `lcp` соседних циклических сдвигов неправильно.

10.11. Как модифицировать алгоритм Касаи для поиска `lcp` соседних в лексикографическом порядке циклических сдвигов строки?

10.12. Дана строка s длины n и перестановка размера n. За время $\mathcal{O}(n)$ проверьте, что перестановка является суффиксным массивом данной строки.

Задачи не по теме

10.13. Докажите, что если для каких-то двух строк $s+t=t+s$, то у s и t есть общий период q, то есть $s=q^i$ и $t=q^j$ для некоторых i и j.

10.14. Предложите способ найти строку длины m в другой строке, используя только $\mathcal{O}(m)$ дополнительной памяти

(a) с помощью префикс-функции

(b) с помощью `z`-функции

10.15. **Строки Фибоначчи**. Определим f_0 = «`b`», f_1 = «`a`», а все следующие f_i как $f_{i-1}+f_{i-2}$.

(a) Докажите, что существует k такое, что для всех $n \geqslant k$ строка f_n^2 является префиксом f_{n+2}.

(b) Докажите, что существует k такое, что для всех $n \geqslant k$ строка $(f_n)_{\to|f_n|-2}$ является палиндромом.

(c) Докажите, что существует k такое, что для всех $n \geqslant k$ строка f_n является бордером строки f_{n+2}.

Баллы

Домашнее задание

★ **Правила сдачи**

- Письменное ДЗ оформляется в LaTeX, написанные от руки решения приниматься не будут
- Решения надо прислать **до 16 декабря 2023, 23:59** на почту `itmo.algo.teaching+y2021@gmail.com`
- Тема письма должна быть указана в виде «`Группа HWBonus Фамилия Имя`» (без кавычек), например: «`M3230 HWBonus Иванов Иван`». Обратите внимание, что буква «M» в названии группы – латинская
- К письму должны быть приложены как сгенерированный `.pdf`-файл, так и исходный `.tex`-файл

1. Дан взвешенный граф на n вершинах, между каждой парой вершин u и v определена функция расстояния $d(u, v)$. За время $\mathcal{O}(n^3)$ определите, какое минимальное расстояние в сумме придется пройти людям, находящимся в вершинах s и t, чтобы поменяться местами, ни разу не оказавшись на расстоянии меньше D друг от друга.
2. Дана табличка $n \times m$, в клетке (i, j) стоит крестик. Играют два игрока, за ход игрок может разрезать табличку по вертикали или горизонтали на две части и выкинуть из игры ту часть, в которой нет крестика. Как определить, какой из игроков имеет выигрышную стратегию
   (a) при $i = j = 1$ (крестик в углу)?
   (b) в произвольном случае?
3. Можно ли придумать хеш для строк такой, что для любой строки s ее хеш совпадает с хешом любого ее циклического сдвига? (Разумеется, вероятность коллизии должна быть $o(1)$). Приведите пример или объясните, почему нельзя.
4. Вспомните автомат КМП. Можно ли аналогичным образом построить z-автомат (автомат для z-функции)? Обоснуйте ответ.
5. Покажите, как найти максимальную подпоследовательность (не обязательно подряд идущих) символов строки длины n, являющуюся палиндромом,
   (a) за время $\mathcal{O}(n^2)$
   (b) за время $o(n^2)$ `[нет гарантий, что этот пункт решается]`
6. **Бесполезные операции на боре**
   (a) Определим *префиксную ссылку* в боре по аналогии с суффиксной ссылкой. Есть ли смысл в такой функции?
   (b) Покажите, как для каждой вершины бора найти *бордерную ссылку* – вершину, задающую строку, являющуюся максимальным бордером данной. Предподсчет $\mathcal{O}(n)$ (количество вершин), ленивое вычисление за $\mathcal{O}(1)$.

Неделя 11. Быстрый цифровой бор

Практика

11.1. **Сортировка Киркпатрика**. На основе идеи дерева ван Эмде Боаса придумайте сортировку n чисел из интервала $[0, 2^w)$, работающую за $\mathcal{O}(n \log w)$.

11.2. Почему дерево ван Эмде Боаса не работает, если минимум и максимум не хранить в вершинах отдельно?

11.3. Почему оптимальное время работы `Y-fast trie` достигается при разбиении чисел на бакеты размера порядка C? Можно ли добиться меньшего времени работы, если строить структуру не в два слоя, а больше?

11.4. Представьте, что при разбиении «вселенной» чисел на кластеры мы используем не $\sqrt{u}$ кластеров по $\sqrt{u}$, а $u^{\frac{1}{k}}$ кластеров по $u^{1-\frac{1}{k}}$. Запишите и решите рекурренты для времени работы операций и занимаемой памяти быстрого цифрового бора.

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

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

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

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

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

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

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

11.12. Дан массив целых чисел. Найдите пару чисел (a_i, a_j) такую, что значение $a_i \oplus a_j$ максимально, за время

   (a) $\mathcal{O}(n \log C)$
   (b) $\mathcal{O}(n \log \log C)$

11.13. 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, за время $\mathcal{O}(n \log C)$.

11.14. 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 неможетсделать ход. Определите, кто выиграет при оптимальной игре обоих игроков, за время $\mathcal{O}(\sum |s_i|)$.

11.15. Задано взвешенное дерево на n вершинах с весами на каждом ребре. Вес каждого пути в дереве не превосходит $42n$. Решите задачу `Weighted LA` (в какую вершину мы попадем, поднявшись из v на d вверх) за $\mathcal{O}(n \log \log n)$ предподсчета и $\mathcal{O}(\log \log n)$ на запрос.

Неделя 12. Суффиксный автомат

Практика

12.1. Постройте суффиксный автомат для строки

(a) «`abbbbbb`»

(b) «`aababca`»

(c) «`aaaaaba`»

(d) «`abcdefa`»

(e) «`abcbcba`»

Рекомендация: это хороший способ разобраться в рассмотренном алгоритме.

12.2. Найдите количество различных подстрок в строке за время $\mathcal{O}(n)$ с помощью суффиксного автомата.

12.3. Научитесь за $\mathcal{O}(n)$ отвечать на запросы «сколько раз подстрока $s_{l:r}$ входит в s» с помощью суффиксного автомата.

12.4. Докажите, что построение суффиксного автомата занимает $\mathcal{O}(n)$ времени:

(a) докажите, что вершин в суффиксном автомате $\mathcal{O}(n)$;

(b) докажите, что ребер вида $p \to q$, где $q.\mathtt{len} = p.\mathtt{len} + 1$ в суффиксном автомате $\mathcal{O}(n)$;

(c) докажите, что ребер в суффиксном автомате $\mathcal{O}(n)$.

Из этого следует, что время построения — $\mathcal{O}(n)$, так как на создание каждой вершины и каждого ребра тратится $\mathcal{O}(1)$ времени.

12.5. Предложите алгоритм построения суффиксного дерева по суффиксному автомату за время $\mathcal{O}(n)$.
Подсказка: структура дерева суффиксных ссылок в автомате может помочь.

12.6. С помощью суффиксного автомата найдите k-ю в лексикографическом порядке подстроку строки s длины n за время $\mathcal{O}(\text{answer})$.

12.7. Даны две строки s_1 и s_2 суммарной длины n. Найдите за время $\mathcal{O}(n)$ максимальную общую подстроку s_1 и s_2.

12.8. Предыдущая задача, но для k строк суммарной длины n.

12.9. Дана строка s длины n. Найдите самую длинную ее подстроку, являющуюся палиндромом, за время $\mathcal{O}(n)$.

12.10. Дана строка s длины n. Найдите самую короткую строку, не являющуюся ее подстрокой, за время $\mathcal{O}(n)$ с помощью суффиксного автомата.

12.11. С помощью суффиксного автомата найдите за $\mathcal{O}(n)$

(a) самую длинную подстроку, имеющую хотя бы два вхождения в данную

(b) самую длинную подстроку, имеющую хотя бы k вхождений в данную

(c) количество подстрок, имеющих хотя бы k вхождений в данную

12.12. Дан набор строк s_i суммарной длины n. За время $\mathcal{O}(n)$ найдите для каждой строки

(a) ее минимальный префикс, не являющийся префиксом никакой другой строки из набора

(b) ее минимальную подстроку, не являющуюся подстрокой никакой другой строки из набора

(или скажите, что таких нет).

12.13. Дана строка s длины n. Требуется за $\mathcal{O}(k \log(n+k))$ отвечать на запросы «даны k подстрок данной строки, найдите две из них, имеющие наибольший общий префикс».

12.14. Дана строка s длины n. Найдите число пар строк (x, y), что x – подстрока y, y – подстрока s и число вхождений x в s равно числу вхождений y в s за время $\mathcal{O}(n)$.