Первый курс, осенний семестр 2020/21
Практика по алгоритмам #2

Всё глубже и глубже
22 января

Собрано 22 января 2021 г. в 12:34

Содержание

Всё глубже и глубже

1. **Самый длинный цикл**

   Найти в неорграфе самый длинный цикл. За сколько можно решать такую задачу?

2. **Эйлеровость**

   a) Дополнить неорграф до Эйлерова минимальным числом ребер. $\mathcal{O}(V+E)$.
   b) Разбить все ребра **неор**графа на минимальное число путей за $\mathcal{O}(V+E)$.
   c) Найти гамильтонов цикл на графе бинарных строк длины n. Ребро между строками есть, если $(n-1)$-суффикс первой строки совпадает с $(n-1)$-префиксом второй строки.
   d) Найти 0-1 строку min длины, содержащую как подстроки все 0-1 строки длины n.

3. **Принцип домино**

   На столе (на плоскости) стоят доминошки. Каждая доминошка стоит на ребре. Вася хочет по очереди уронить все доминошки на стол, каждую он может ронять в любую из двух сторон. Может ли Вася выбрать такой порядок и направления падения, чтобы ни в какой момент времени доминошки не касались друг друга?

4. **2-List-Coloring**

   2-List-Coloring: есть граф и k цветов, для каждой вершины дан список из двух цветов, в которые ее можно красить. Найти правильную покраску или сказать, что такой нет. $\mathcal{O}(V+E)$.

5. **3-List-Coloring**

   3-List-Coloring: есть граф и k цветов, но для каждой вершины дан список из трех цветов, в которые ее можно красить. Найти правильную покраску или сказать, что такой нет. $\mathcal{O}(1.5^V E)$.

6. **3-связность**

   Проверить граф на 3-связность

   a) вершинная за $\mathcal{O}(VE)$
   b) рёберная за $\mathcal{O}(VE)$

7. **Достижимость**

   Дан произвольный граф, $N, M \leqslant 10^5$.

   a) найти количество пар вершин $\langle a, b\rangle$: из a достижима b.
   b) ...и $K \leqslant 10^5$ offline-запросов «достижима ли b из a».

8. **Цикл длины 3**

   Дан неорграф. Найти в нём цикл длины ровно 3.

   a) $\mathcal{O}(V^3)$
   b) $\mathcal{O}(VE)$
   c) $\mathcal{O}(VE/w)$
   d) $\mathcal{O}(E\sqrt{E})$ (есть два решения!)
   e) За $\mathcal{O}(V^{3-\varepsilon})$ при $E = \Theta(V^2)$

9. **(*) Цикл длины 4**

   Дан неорграф. Найти в нём цикл длины ровно 4 за $\mathcal{O}(V^2)$.

10. **(∗) Взвешенное дополнение до эйлеровости**

    У каждой вершины есть вес. Вес любого ребра – сумма весов концов. Дан неорграф, дополнить его до Эйлерова, добавив ребра минимального суммарного веса.

11. **(∗) Центры дерева**

    Дано дерево. Выбрать на нем k вершин так, чтобы сумма расстояний до ближайшей из выбранных вершин была минимальна.

Разбор задач практики

1. **Самый длинный цикл**

   Если есть гамильтонов цикл, он и есть ответ. Так что мало шансов решить такую задачу за полином. Умеем за $\mathcal{O}(2^n n)$.

2. **Эйлеровость**

   a) **Дополнить неорграф до Эйлерова.**
   Если граф связен – любым образом паросочетаем нечётные вершины.
   Если несвязен, выписываем компоненты связности по циклу и сперва добиваем связности, добавляя рёбра между соседними компонентами. Если какая-то компонента содержит только чётные вершины, в какую-нибудь вершину проведём два ребра.

   b) **Разбить ребра на минимальное число путей.**
   Все компоненты связности обработаем независимо. Дополним связный граф до Эйлерова, найдём Эйлеров цикл, удалим добавленные ранее рёбра, цикл распался на пути.
   Путей будет столько же, сколькими ребрами дополнили до Эйлерова. Наоборот, имея пути, можно их по циклу соединить и получить Эйлеров. Так что минимизация этих задач эквивалентна.

   c) **Найти гамильтонов цикл на графе 0-1 строк длины n.**
   Гамильтонов не умеем искать быстро. Умеем Эйлеров.
   Построим граф, в котором 0-1 строки длины n – рёбра. Вершинами будут строки длины $n-1$. Ребро соединяет свой суффикс длины $n-1$ со своим префиксом длины $n-1$.
   Граф ориентированный. Ребро: префикс $\to$ суффикс. У каждой вершины входящая степень $=$ исходящей $= 2 \Rightarrow \exists$ Эйлеров цикл.

   d) **0-1 строка** $\min$ **длины, содержащая как подстроки все 0-1 строки длины n.**
   Пишем строку из $n-1$ нуля, затем берем граф из прошлого пункта и пишем по символу с ребер в порядке Эйлерова обхода, стартующего с вершины из $n-1$ нуля. Тогда все подстроки длины n будут разными, длина 2^n+n-1.

3. **Принцип домино**

   Задача сводится к `2-SAT`. x_i – в какую сторону падает i-я доминошка.

4. **2-List-Coloring**

   Задача сводится к `2-SAT`. У вершины v есть список c_v из двух цветов. Переменная $x_v = b$ означает «v покрашена в цвет $c_v[b]$».

   Если есть ребро $\{v, u\}$, то вершины v и u не могут быть покрашены в один цвет. Если $c_v[0] = c_u[0]$, записываем условие $x_v = 1 \vee x_u = 1$. И так в каждом из четырёх случаев:
   `forn(i, 2) forn(j, 2) if (c[v,i] == c[u,j]) ....`

   Если исходный граф состоял из V вершин и E рёбер, мы получили `2-SAT` из V переменных и $\leqslant 2E$ клозов.

   Напомним, что для `2-SAT` из n переменных и m клозов строится орграф из $2n$ вершин и $2m$ ребер, в котором за $\mathcal{O}(n+m)$ ищутся компоненты сильной связности.

5. **3-List-Coloring**

   Если запретить случайный цвет в каждой вершине, то получится `2-List-Coloring`, который решается за $\mathcal{O}(E)$.

Если есть правильная покраска, то у каждой вершины мы запретили ее цвет с вероятностью $\leqslant \frac{1}{3}$. С вероятностью $\geqslant (\frac{2}{3})^V = p$ у всех вершин не запрещен нужный цвет.

Получили решение за $\mathcal{O}(E)$, которое работает с вероятностью не менее p. Если повторить его $\frac{1}{p}$ раз, вероятность неудачи будет не более $(1-p)^{1/p} \approx e^{-1}$. Если повторить его $20\frac{1}{p} = \mathcal{O}(1.5^V)$ раз, вероятность неудачи будет не более $e^{-20} \approx 2 \cdot 10^{-9}$.

6. **3-связность**

   a) Вершинная. Нужно узнать, можно ли удалением двух вершин разрушить связность. Одну переберем, удалим, на оставшемся графе за $\mathcal{O}(E)$ проверяем наличие точек сочленения.
   b) Рёберная. Можно так же перебрать, выйдет $\mathcal{O}(E^2)$. Но ясно, что для разрушения связности нужно удалить хоть одно ребро из остовного дерева. Так что перебираем только $\mathcal{O}(V)$ ребер.

7. **Достижимость**

   Сначала надо сконденсировать граф. У нас была динамика: маска достижимых из a – это `OR` масок детей. Но это $V^2 = 10^{10}$ *бит* памяти, довольно много.

   Разобьём динамику на $\frac{V}{w}$ частей, в каждой будем считать достижимость w вершин из остальных. Время то же: $\frac{V}{w} \cdot E = \frac{VE}{w}$, но дополнительной памяти уже всего $\mathcal{O}(V)$.

   На запросы отвечаем offline. Найдя достижимость k вершин, ответили на все запросы про пути из любой вершины в какую-то из этих k.

8. **Цикл длины 3**

   a) $\mathcal{O}(V^3)$: переберем все тройки вершин, проверим, что треугольник.
   b) $\mathcal{O}(VE)$: переберем ребро (z, x) и вершину y, проверим, что есть ребра $(x, y), (y, z)$.
   c) $\mathcal{O}(VE/w)$: переберем ребро (z, x). Вершина y ищется в пересечении `bitset`-ов соседей z и x.
   d) $\mathcal{O}(E\sqrt{E})$. Сначала, как в предыдущем пункте, переберем вершину x и ребро (y, z), но только при $\deg(x) > \sqrt{E}$. Таких x в графе не более $2\sqrt{E}$. Так нашли все треугольники, содержащие хоть одну вершину большой степени, за $\mathcal{O}(E\sqrt{E})$.

      Теперь найдем треугольники, в которых все вершины маленькой степени. Перебирем ребро (y, z), перебираем x – соседа z, смотрим, есть ли ребро (x, y). Если все степени x, y, x не более $\sqrt{E}$, мы нашли новый треугольник.

      Вместо этого можно перебрать ребро (z, x), перебрать y среди соседей вершины меньшей степени x, z, проверить наличие ребра во вторую. Сложность $\sum_{(z,x)} \min(\deg(x), \deg(z))$. Утверждается, что это $\mathcal{O}(E\sqrt{E})$.
   e) $o(V^3)$. Если возвести матрицу смежности графа в стпень k, то в ячейке $[u, v]$ будет число путей $u \rightsquigarrow v$ длины k. Пути не обязательно простые, могут повторяться вершин и ребра, но путь длины 3 из v в v – всегда треугольник.

      То есть надо посчитать A^3 за два умножения матриц, матрицы умеют умножать за $\mathcal{O}(n^{2.37})$. Потом за $\mathcal{O}(n^2)$ проверяем, что для какой-нибудь v $A^3[v, v] > 0$.

9. **Цикл длины 4**

   Переберем пары ребер с общим концом (v, u), (v, w), пометим пару $\{u, w\}$, как имеющую общего соседа. Если нашли уже помеченную пару, то у этой пары два общих соседа, это цикл длины 4. Поскольку мы остановимся, как только хоть одна пара повторилась, время работы не больше, чем число различных пар, $\mathcal{O}(V^2)$.

10. **Взвешенное дополнение до эйлеровости**

    Аналогично невзвешенному случаю, соединяем циклом нечетные вершины, к ним в любом случае нужно добавить ребро.

    Если есть компонента без нечетных вершин, возьмем вершину min; веса и добавим ей свободную степень два, к этой компоненте в любом случае пришлось бы провести два ребра.

11. **Центры дерева**

    `f[v,k,±d]` – минимальная сумма расстояний в поддереве вершины `v`, если выбрано ровно `k` вершин, а расстояние до ближайшей выбранной или $|d|$ вниз (для $d > 0$), или $|d|$ вверх (для $d < 0$, как бы смотрим в будущее). Состояний $\mathcal{O}(n^3)$, динамика работает за $\mathcal{O}(n^4)$.

    При этом есть тривиальное решение за $\mathcal{O}(2^n n)$, которое вполне работает для $n \leqslant 20$.

Домашнее задание

3.1. Обязательная часть

1. **(2) Разноцветные двери**

   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 $\mathcal{O}(\texttt{polynom}(V, E))$, **(+1)** $\mathcal{O}(V+E)$.

   Подсказка: как бы вы решили задачу для Эйлерова графа?

2. **(2) Разбиение на пути**

   Разбить все рёбра ориентированного графа на минимальное число путей.

3. **(2) Разбиение на паросочетания**

   Дан 2^k-регулярный двудольный граф (степени всех вершин равны 2^k). Нужно разбить все его ребра на **совершенные** паросочетания за $\mathcal{O}(kE)$.
   Совершенное паросочетание – 1-регулярный граф.

4. **(2) Необходимые для достижимости ребра**

   Дан неорграф.
   Найти все рёбра, которые обязательно будут лежать на пути от a до b. $\mathcal{O}(V+E)$.

5. **(3) Необходимые для достижимости вершины**

   Дан неорграф.
   Найти все вершины, которые обязательно будут лежать на пути от a до b. $\mathcal{O}(V+E)$.

6. **(3) Провода**

   Есть n проводов. Есть круг из $2n$ разъемов, каждый разъем имеет тип от 1 до n (соответствующий номеру провода, который можно туда воткнуть), каждый тип встречается ровно два раза. У каждого провода есть цвет. В целях безопасности нельзя втыкать провода одинакового цвета в соседние разъемы. Найти способ соединить каждый из n проводов с одним из двух подходящих разъемов, не нарушив правила на цвета соседних.

3.2. Дополнительная часть

1. **(3) Необходимые для достижимости вершины в орграфе**

   Дан **орграф**.
   Найти все вершины, которые обязательно будут лежать на пути от a до b. $\mathcal{O}(V+E)$.

   В этой задаче без строгого доказательства можно получить максимум половино от полного балла.

2. **(4) Удаление двух ребер**

   Нужно найти количество способов удалить из связного неорграфа два ребра так, чтобы он перестал быть связным.

   a) **(2)** $\mathcal{O}(V^2)$
   b) **(2)** $\mathcal{O}(E)$

3. **(3) Центры дерева**

   Дано дерево. Выбрать на нем k вершин так, чтобы максимальное расстояние от вершин дерева до ближайшей из выбранных вершин было минимально. $\mathcal{O}(n \log n)$.

4. **(3) Центры окружности**

   По кругу стоят n точек. Выбрать k из них так, чтобы минимальное расстояние между соседними выбранными было максимально. $\mathcal{O}(n \cdot \texttt{polylog})$.