Первый курс, осенний семестр 2020/21
Практика по алгоритмам #1

Поиск в глубину

15 января

Собрано 15 января 2021 г. в 12:20

Содержание

1. Поиск в глубину **1**

2. Разбор задач практики **3**

3. Домашнее задание **5**

3.1. Обязательная часть 5

3.2. Дополнительная часть 6

Поиск в глубину

1. **Нечётный цикл**

   За $\mathcal{O}(V{+}E)$ найти в неорграфе цикл нечетной длины.

2. **Две клики**

   Есть n человек, между ними есть симметричное отношение дружбы. Разбейте человек на две группы, чтобы в каждой группе все друг с другом попарно дружили.

3. **Разбиение на дружелюбные доли**

   У каждой вершины не более 3 врагов.
   Разбить на 2 доли так, чтобы с вершиной в долю попало не более 1 врага. $\mathcal{O}(V{+}E)$.

4. **Транзитивное замыкание**

   Дан орграф, построить матрицу достижимости. $V \leqslant 20\,000$, $E \leqslant 200\,000$.

5. **Кластеризация на два кластера.**

   Даны объекты, и матрица расстояний d_{ij} (непохожести объектов). Нужно разбить объекты на два множества так, чтобы максимальный из диаметров множеств был минимален.

   Пример объектов: точки на плоскости.
   Пример объектов: тексты и их расстояние Левенштейна.

6. **Компактификация**

   Придумать способ хранения графа, чтобы памяти он занимал $\mathcal{O}(m)$, а dfs с ним работал за $\mathcal{O}((n+m) \cdot poly(\log n))$.

7. **В мире перестановок**

   Даны перестановки p и q из 10 элементов. Можно ли в каком-то порядке применяя к тождественной перестановки p и q, получить перестановку z?

8. **Снятие пометок и циклы**

   Вася хочет перебрать все-все-все циклы в неорграфе. Для этого он придумал «запустить обычный dfs для поиска цикла и снимать при выходе из dfs пометки». За сколько в худшем работает такой алгоритм? Какие у него проблемы? Как всё-таки посчитать число различных циклов?

9. **Кактус, сэр!**

   Граф называется рёберным кактусом, если каждое ребро лежит ровно на одном цикле. Дан граф, проверьте, является ли он кактусом, и, если да, найдите все его циклы. $\mathcal{O}(n+m)$.

10. **Строкапуты**

    Дан орграф, на рёбрах написаны буквы. Найдите путь в орграфе, на котором написана ровно данная строка s. За полином.

11. **(*) Здоровенная зебра**

    Дана матрица из чисел. Выберите наибольшее связное по стороне множество клеток, чтобы было использовано не более двух различных чисел. $\mathcal{O}(wh)$.

12. **(*) Максимальное число различных циклов**

    Есть неорграф, нужно выбрать максимальное число простых циклов так, чтобы каждый следующий содержал хотя бы одно новое ребро. $\mathcal{O}(E)$.

13. **(*) Дополнение слабосвязного графа до сильносвязного**

    a) Перейдем к конденсации.
    b) Заметим стоки и истоки, построим новый граф.
    c) Угадаем ответ
    d) Идея #1: попытаемся провести ребро, которое на 1 уменьшает число истоков и стоков.
    e) Идея #2: попробуем замкнуть в цикл любое max по включению паросочетание.
    f) Идея #3: $\mathcal{O}(V+E)$. Возьмем любой максимальный по включению непересекающийся по вершинам набор путей из истоков в стоки.

Разбор задач практики

1. **Нечётный цикл**

   Красим граф в два цвета. Если видим ребро, ведущее из текущую вершину в уже покрашенной в тот же цвет – нашли нечетный цикл, он лежит на стеке рекурсии.

2. **Разделение на две клики**

   Инвертируем все ребра: если между парой вершин нет ребра, добавим, иначе уберем. `g[i,j] ^= 1`. Теперь надо разбить на два независимых множества $\Leftrightarrow$ покрасить в два цвета.

   Важное замечание: это работает за $\mathcal{O}(V^2)$, в инвертированном графе ребер $V^2 - E$.

3. **Разбиение на дружелюбные доли**

   Разобьем как-нибудь. Метод локальных оптимизаций: если у какой-то вершины в её доле 2 врага, перекинем её в другую долю.

   Каждый раз уменьшается суммарное число рёбер-врагов внутри долей, поэтому надо $\mathcal{O}(E)$ шагов. Чтобы быстро находить плохие вершины, сделаем это dfs-ом.

   ```
   void dfs(int v) {
           if (bad(v)) {
                   color[v] ^= 1;
                   for (int x : g[v]) dfs(x);
           }
   }
   for (int v = 0; v < n; v++)
           dfs(v);
   ```

   Заметим, что тут нет пометок посещенных вершин, но рекурсивные вызовы возникают только после перекидывания вершины, поэтому время $\mathcal{O}(V + E)$.

4. **Транзитивное замыкание**

   Сконденсируем граф. Из каждой вершины достижима её ксс и ещё какие-то. Какие?

   Динамика по конденсации `bitset dp[v]` – множество вершин, достижимых из v. `dp[v]` – `OR` bitset'ов детей v. Динамика ленивая (dfs-о-динамика).

5. **Кластеризация на два кластера.**

   Бинпоиск по ответу.
   Внутри если $d_{ij} > x$ нужно класть i и j в разные части $\Rightarrow$ раскраска в два цвета.

   Более продвинутое решение: сортируем рёбра по возрастанию, добавляем их в таком порядке и СНМ-ом поддерживаем двудольность.

6. **Компактификация**

   Сортированный список рёбер (a, b).
   Перебор рёбер из a: бинпоиском найти первое a и за степень a перебрать рёбра.
   Время `dfs` – $\mathcal{O}(n \log m + m)$. Памяти нужно $8m$ для орграфа $16m$ для неорграфа.

7. **В мире перестановок**

   Видим граф из $10! < 4\,000\,000$ вершин, на нём dfs.

8. **Снятие пометок и циклы**

   Работает за $n!$ на полном графе.

   Проблема = каждый цикл найдётся несколько раз (разный предпериод, 2 направления прохождения, вход в разные вершины).

   Решение проблемы = после нахождения цикла выберем переберём циклический сдвиг и направление, выберем пару с минимальным `vector<int>` цикла, сложим всё это добро в `set`.

9. **Кактус, сэр!**

   Обычный dfs по неорграфу, который ищет циклы, при нахождении цикла помечает все рёбра. Если какое-то ребро хочет пометиться дважды, не кактус, брякаемся.

10. **Строкапуты**

    Динамика `dp[v,i]` – можно ли оказаться в вершине v, выписав первые i символов s.

11. **(*) Здоровенная зебра**

    Сожмём компоненты одного цвета. В новом графе у каждой вершины есть вес (размер компоненты) и цвет (число из матрицы). Тип ребра в сжатом графе = типы его концов. Для каждого типа возьмём все рёбра такого типа и `dfs` по ним. Только аккуратно, чтобы получилось за $\mathcal{O}(E)$. Для этого нужно ходить только по вершинам, из которых есть рёбра.

12. **(*) Максимально число различных циклов**

    Рассмотрим любое остовное дерева `dfs`. Его рёбра нужны, чтобы хотя бы посетить вершины. А все не древесные рёбра образуют циклы, всего $m - (n{-}1)$ циклов.

    Кстати, полученные циклы образуют базис в «пространстве циклов».

13. **Дополнение связного графа до сильносвязного**

    Перешла в ДЗ. Разбор там же

Домашнее задание

3.1. Обязательная часть

1. **(2) Покраска в 4 цвета**

   За $\mathcal{O}^*(2^n)$ проверить, можно ли покрасить граф в 4 цвета.
   Подсказка: в два цвета красить очень просто.

2. **(2) Минимизация максимального ребра на цикле**

   Дан неорграф с целыми положительными весами на рёбрах. Найти в неорграфе такой цикл, что максимальный вес ребра этого цикла минимален. $\mathcal{O}((V+E)\log E)$.
   Решить, используя только известное нам.

3. **(2) Ещё один вариант разбиения на две доли**

   Разбить множество вершин неорграфа на две доли так, чтобы у каждой вершины был сосед в другой доле. $\mathcal{O}(V+E)$.

4. **(3) Ромбики**

   Найти в неорграфе «ромбик с диагональю»: 4 вершины a, b, c, d с 5 рёбрами ab, bc, cd, da, ac.
   Оценка: (1) $\mathcal{O}(V^3)$, (1) $\mathcal{O}(VE)$, (1) $\mathcal{O}(\frac{VE}{w})$, (+1) $\mathcal{O}(E^{3/2})$.

5. **(3) Покраска в 3 цвета**

   Дан изначально пустой неорграф. В него добавляются рёбра и вершины. При этом поддерживается инвариант, что степени всех вершин не более 5. Поддерживать покраску вершин в 3 цвета так, чтобы у каждой было не более одного соседа того же цвета. При добавлении новых ребра/вершины обновлять покраску за амортизированное $\mathcal{O}(1)$.

6. **(2) Путь через ад**

   Дан орграф. В некоторых вершинах живут монстры. Монстры бывают трёх типов. Проходя через некоторые вершины, можно получить иммунитет к некоторым типам монстров (три разных иммунитета). В каждый момент времени каждый из трёх иммунитетов или есть, или нет, копить их нельзя. Иммунитет при встрече с монстром спасает от гибели ровно один раз и пропадает. Можно ли из вершины a дойти до вершины b и не умереть? ($V, E \leqslant 10^5$).

3.2. Дополнительная часть

1. **(3) Минимизация максимальной исходящей степени**

   Дан неорграф. Ориентировать его так, что максимальная исходящая степень была минимальна. $\mathcal{O}(E^2)$. Можно еще на полилог.

2. **(3) Сильная связность**

   Сильная связность (задача с доп практики). Решение за $\mathcal{O}(V+E)$: построим конденсацию, выделим стоки и истоки. Затем найдем за $\mathcal{O}(V+E)$ максимальное по включению множество непересекающихся по вершинам путей из истоков в стоки. То есть, не существует еще одного пути из истоков в стоки, не пересекающегося с уже выбранным. Пусть i-й путь начинается в a_i, заканчивается в b_i, всего путей k. Добавим ребра $(b_1, a_2), (b_2, a_3), \ldots, (b_n, a_1)$. Это неполное описание решения. Задача в том, чтобы довести его до конца, доказать корректность, доказать линейность времени работы.

3. **(3) Существование пути**

   Напишите детерминированное решение, которое имеет доступ к оракулу $g(i, j)$ – есть ли ребро между i и j, и должно, используя полилогарифм от n памяти проверить наличие пути из a в b в графе из n вершин.