SPb HSE, 1 курс, зима 2024/25
Практика по алгоритмам #14

dfs-2

14 января

Собрано 15 января 2025 г. в 13:11

Содержание

dfs-2

1. **Crazy horses**

   На шахматной доске стоят 2 коня и ещё какие-то фигуры. Нужно, ходя конями по очереди, поменять их местами. В каждой клетке в каждый момент может стоять только одна фигура.

2. **Сложности плюсов**

   Замените знаки вопросов на куски кода так, чтобы всё было хорошо.

```
struct Edge {
    int vertex, weight;
};
vector<vector<Edge>> g(n);
vector<int> used(n);
function<???> dfs = [???](int v) {
    used[v] = 1;
    for (??? : g[v])
        if (!used[x])
            dfs(x);
};
dfs(0);
```

3. **Нечётный цикл**

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

4. **Две клики**

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

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. **Три непростые клики**

   Покажите, что задача «проверить, можно ли вершины графа разбить на три клики» трудная.

7. **Единственность топсорта**

   Дан ацикличный орграф, проверить единственность топсорта за $\mathcal{O}(V+E)$.

8. **Ацикличная ориентация**

   Ориентировать неорграф так, чтобы он стал ацикличным, за $\mathcal{O}(V+E)$.

9. **Сильносвязная ориентация**

   Ориентировать неорграф так, чтобы он стал сильносвязным, за $\mathcal{O}(V+E)$.

10. **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 строку из 0 и 1 минимальной длины, которая содержит все строки длины n, как подстроки.

11. **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».

12. (*) **Четный цикл**

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

13. (*) **Минимизация максимального ребра в пути**

    Найдите путь из s в t: максимальное ребро минимально. $\mathcal{O}(E)$.

14. (*) **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. **Crazy horses**

   Вершина нового графа $\langle x_1, y_1, x_2, y_2, who \rangle$, где who – кто ходит.

2. **Сложности плюсов**

   ```
   function<void(int)> dfs = [&](int v)
   for (auto [x, _] : g[v])
   ```

3. **Нечётный цикл**

   Красим граф в два цвета. Если видим ребро из текущей, ведущее в вершину такого же цвета – нашли нечетный цикл. Если цикл мы искали dfs-ом, он как раз лежит на стеке рекурсии.

4. **Разделение на две клики**

   Если человека v поместили в первое множество, то всех его не друзей обязательно поместить во второе $\Rightarrow$ такой же алгоритм, как покраска в два цвета.

   Также можно заметить, что мы делим на две клики, а покраска = поделить на две антиклики $\Rightarrow$ Инвертируем все ребра: если между парой вершин нет ребра, добавим, иначе уберем. `g[i,j] ^= 1`. И покрасим в два цвета. *Замечание*: работает за $\mathcal{O}(V^2)$, так как в инвертированном графе ребер $V^2 - E$.

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. **Три непростые клики**

   В три цвета покрасить быстро не умеем (знаем с лекции) $\Rightarrow$ и на три клики разделить не умели (если бы умели, применили бы это к задаче про покраску).

7. **Единственность топсорта**

   Делаем topsort, проверяем, что из каждой вершины есть ребро в следующую =)

   И правда, если между какими-то соседями нет ребра, их можно поменять местами.

8. **Ацикличная ориентация**

   Ориентируем ребра от меньшей вершины к большей.

   Способ #2. Запустим dfs, ориентируем все ребра вниз, то есть ребра дерева и обратные из предка в потомка.

   В обоих случаях цикла нет, ибо на любом пути либо строго растет номер вершины, либо глубина в дереве dfs.

9. **Сильносвязная ориентация**

   Ориентируем ребра так, как на них впервые посмотрит dfs.

   Получится, что ребра дерева dfs будут ориентированы вниз, а обратные вверх.

   Из корня достижимо всё. Откуда угодно можно попасть в корень, потому что откуда угодно можно подняться выше: если нельзя, то между поддеревом вершины и тем, что над ней, есть только одно ребро, тогда ориентация была исходно невозможна.

10. **Дополнение связного графа до сильносвязного**

    Перешла в ДЗ. Разбор там же.

11. **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 $2^n + n - 1$.

12. **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.

13. **(*) Четный цикл**

    ?

14. **(*) Минимизация максимального ребра в пути**

    ?

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

    Ушла в дз.

Домашнее задание

3.1. Обязательная часть

1. **(2) Разбиение на пути**

   Разбить все рёбра ориентированного графа на минимальное число путей.

2. **(2) Дополнение до сильной связности**

   Вам дан орграф. Проверьте за $\mathcal{O}(V+E)$, можно ли добавить в него одно ребро, чтобы он стал сильносвязным.

3. **(2) Покраска в 4 цвета**

   За $\mathcal{O}^*(2^n)$ проверить, можно ли покрасить граф в 4 цвета.
   Обозначение $\mathcal{O}^*(\dots)$ используется, когда полиномиальная часть времени нам не важна.

   Подсказка: в два цвета красить очень просто.

4. **(2) Mr.Euler и разбиение на паросочетания**

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

3.2. Дополнительная часть

1. **(3) Центры дерева**

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

2. **(2) Дырявый океан**

   Поверхность планеты «dfsland» – один большой океан. В этом океане есть острова, в них есть озёра, в озёрах тоже есть острова, в которых тоже озёра... Жители планеты составили карту местности в виде матрицы $w \times h$, где суша обозначена «`#`», а вода обозначена «`.`», при этом края матрицы обязательно являются водой. Остров – набор клеток суши, связный по стороне. Острова не касаются по углу. Найдите самую длинную по вложенности цепочку островов-озёр. $w, h \leqslant 2\,000$.

3. **(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 $(b_1, a_2), (b_2, a_3), \dots, (b_k, a_1)$. Это неполное описание решения. Задача в том, чтобы довести его до конца, доказать корректность, доказать линейность времени работы.