SPb HSE, 1 курс, осень 2021/22
Практика по алгоритмам #16

Поиск в глубину #2
21 января

Собрано 20 января 2022 г. в 21:37

Содержание

Поиск в глубину #2

1. **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.

2. **Компактификация**

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

3. **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?

4. **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?

5. **Кактус, сэр!**

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

6. **Строкапуты**

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

7. **Самый длинный цикл**

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

8. **3-связность**

   Проверить граф на 3-связность

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

9. **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».

10. **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 $E = \Theta(V^2)$

11. **(*) Цикл длины 4**

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

12. **(*) Здоровенная зебра**

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

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

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

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. **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.

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]` – это `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. Важно запомнить, что dfs быстро ищёт «один какой-то цикл», а вот «найти все циклы» – это уже экспонента.

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

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

9. **Кактус, сэр!**

   Решение: dfs по неорграфу, который ищет циклы, а при нахождении цикла помечает все найденного цикла рёбра. Если какое-то ребро хочет пометиться дважды, это не рёберный кактус, брякаемся. Каждое ребро помечаем один раз $\Rightarrow \mathcal{O}(V+E)$.

   P.S. Чтобы проверять «вершинный кактус», действуем также, но помечаем вершины.

10. **Строкапуты**

    Динамика `dp[v,i]` – можно ли оказаться в вершине v, выписав первые i символов строки s. Начальное состояние $\forall v$ `dp[`v`,0] = true`. Конец пути – любая v: `dp[`v`,`$|s|$`] = true`.

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 запустим `dfs` по ним.
    Чтобы суммарное время работы было линейным `dfs` по должен работать ровно за $\mathcal{O}(E)$ – обнулять пометки за $\mathcal{O}(1)$ и ходить только по вершинам, из которых $\exists$ 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) Необходимые для достижимости ребра**

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

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

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

3. **(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? ($V, E \leqslant 10^5$).

4. **(3) 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.

   За $\mathcal{O}(wh)$ найдите:

   a) количество островов у которых внутри есть другие острова,
   b) остров с самым большим количеством островов внутри,
   c) самую длинную по вложенности цепочку островов-озёр.

5. **(2+3) Удаление двух ребер**

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

   a) **(1)** $\mathcal{O}(E^2)$
   b) **(1)** $\mathcal{O}(VE)$
   c) **(2)** $\mathcal{O}(V^2)$
   d) **(1)** $\mathcal{O}(E)$

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. **(2+2) Динамическая сильная связность**

   В орграф добавляются рёбра.
   Нужно поддерживать количество компонент сильной связности.

   a) $V \leqslant 1\,000, E \leqslant 1\,000\,000, q \leqslant 1\,000$.
   b) $V \leqslant 1\,000, E \leqslant 1\,000\,000, q \leqslant 1\,000\,000$.