2 курс ПМИ, осень 2024/25
Практика по алгоритмам #4

Потоки посложнее
30 сентября

Собрано 30 сентября 2024 г. в 12:44

Содержание

Потоки посложнее

1. **Единственность максимального потока**

   Дан поток из s в t размера k.
   За $\mathcal{O}(E)$ проверить, $\exists$ ли отличный от него поток из s в t размера k?

2. **Котята, девочки, собачки**

   Вспоним задачу из последнего дз. За сколько и почему отработает алгоритм Диница?

3. **Многосочетание**

   Дан двудольный граф. Каждой вершине сопоставлено число a_i.
   Выбрать max число рёбер так, чтобы степени вершин были $\leqslant a_i$.

   За сколько будет работать алгоритм Диница в данном случае?

4. **Глобальный разрез**

   В неорграфе без кратных рёбер удалить min число рёбер так, чтобы увеличилось число компонент связности. $\mathcal{O}(V \cdot Flow)$. За сколько работает Форд-Фалекрсон? В каком случае это быстрее чем Каргер-Штейн?

5. **Взвешенное удаление графа**

   Дан орграф. За одно действие можно удалить все входящие в вершину i рёбра за стоимость $a_i \geqslant 0$ или все исходящие за стоимость $b_i \geqslant 0$. Удалить все рёбра за min стоимость.

6. **Давайте придумаем LR-поток**

   a) Несколько истоков, стоков. Пустить max поток.
   b) А если мы хотим k_i путей, причём из s_i именно в t_i?
   c) Даны несколько заводов (производит a_i товара) и магазинов (нуждается в b_j товара) и дорожная сеть. Придумать план перевозок, который удовлетворит все магазины.
   d) Найти любую LR-циркуляцию ($l_e \leqslant f_e \leqslant c_e$).
   e) Найти любой LR-поток из s в t ($l_e \leqslant f_e \leqslant c_e$).
   f) Найти максимальный LR-поток из s в t ($l_e \leqslant f_e \leqslant c_e$).

7. **Нарушение связности**

   Дан орграф. У каждого ребра есть неотрицательная стоимость удаления.
   Удалить рёбра минимальной суммарной стоимости так, чтобы из s не было пути в t.

8. **Оптимизируем КарШтейна**

   Научитесь одну фазу алгоритма за $\mathcal{O}(V^4)$ делать не за $\mathcal{O}(V^2)$, а за $\mathcal{O}(E\alpha)$ ($c_e \equiv 1$).

9. **Варьируем константы КарШтейна**

   В Каргере-Штейне мы делимся на несколько веток в момент, когда вероятность ошибки $= \frac{1}{2}$.
   Делимся ровно на 2 ветки, и в итоге получаем вероятность ошибки $q = 1 - \frac{1}{\log n}$. А если мы

   a) хотим сделать вероятность ошибки $q = 0.75$, сколько рекурсивных вызовов сделать?
   b) хотим сделать вероятность ошибки $q = 1-\varepsilon$, сколько рекурсивных вызовов сделать?
   c) Как делать дробное число вызовов?

10. **(∗) Оптимизированный поток**

    Научитесь поток размера $|f|$ искать за $\mathcal{O}(|f| \cdot V^2/w)$.

11. **(∗) Малхотра-Кумар-Махешвари**

    Идея для оптимизации Диница: давайте научимся за $\approx\mathcal{O}(V)$ насыщать самую «узкую» вершину в нашем слоистом графе. Придумайте поток за $\mathcal{O}(V^3)$.

Разбор задач практики

1. **Единственность максимального потока**

   Кроме нашего потока f есть другой поток f^* iff
   $\exists$ циркуляция $f^* - f$ в остаточной сети $c - f \Leftrightarrow \exists$ цикл в остаточной сети $c - f$.

2. **Котята, девочки, собачки**

   $c_v \leqslant deg_v \Rightarrow \sum c_v \leqslant 2E \Rightarrow$ фаз Диница не более $\sqrt{E}$.
   Каждая фаза работает за E. Итого $E\sqrt{E}$.

3. **Многосочетание**

   Добавляем рёбра из истока в левую долю, из правой доли в сток.
   На добавленных рёбрах пропускные способности a_i, на рёбрах исходного графа 1.
   Ищем максимальный поток, ответ – насыщенные ребра исходного графа.

   Диниц за $\mathcal{O}(E\sqrt{E})$ по первой теореме Карзанова.

4. **Глобальный разрез**

   Вершина 1 лежит в какой-то половине разреза, нужно найти вершину, лежащую в другой.
   Фиксируем любой исток и перебираем сток. Пустили поток, нашли разрез. Время $\mathcal{O}(V \cdot Flow)$.
   Размер i-го потока не более $\deg_i \Rightarrow$ суммарный размер всех потоков не более E
   $\Rightarrow$ даже обычный Форд-Фалкерсон даст время $\mathcal{O}(E^2)$.

5. **Взвешенное удаление графа**

   Раздвоили вершины, ребро (u, v) перешло в (u_L, v_R).
   Вес u_L равен b_v, вес v_R равен a_v, ищем минимальное по весу вершинное покрытие.

6. **Давайте придумаем LR-поток**

   a) *Несколько истоков, стоков. Пустить max поток.*
   Добавим общий исток и общий сток. Пропускные способности $= +\infty$.

   b) *k_i путей, из s_i именно в t_i*
   Это NP-трудно.

   c) *Транспортная задача (избытки и недостатки).*
   Добавим общий исток и общий сток. Пропускные способности $= a_i$, b_j.

   d) *Найти любую LR-циркуляцию ($l_e \leqslant f_e \leqslant c_e$).*
   1. $\forall$ ребра e пустим l_e единиц потока.
   Для ребра $e\colon a \to b$ образуется недостаток l_e в a и избыток l_e в b.

   2. В каждой вершине сложили избытки и недостатки.
   Получили предыдущую задачу на графе с пропускными способностями $r_e - l_e$.

   e) *Найти любой LR-поток из s в t ($l_e \leqslant f_e \leqslant c_e$).*
   Добавим ещё и ребро $t \to s$ пропускной способности $+\infty$.

   f) *Найти максимальный LR-поток из s в t ($l_e \leqslant f_e \leqslant c_e$).*
   Максимальный LR-поток – f^*, мы уже нашли какой-то LR-поток f.
   Заметим, что $f^* - f$ – максимальный (уже не LR, а обычный) поток в G_f.

7. **Нарушение связности**

   Нужно просто найти min разрез в графе, где c_e = вес ребра.

8. **Оптимизируем КарШтейна**

   Сделаем `random_shuffle` рёбер и будем их добавлять в таком порядке, пока в графе не останется 2 компоненты связности. СНМ даёт времени $\mathcal{O}(m\alpha)$ либо $\mathcal{O}(m + n\log n)$.

9. **Варьируем константы КарШтейна**

   q – вероятность ошибки $\Rightarrow q = \frac{1}{2} + \frac{1}{2}q^k$.

   a) $q = 0.75 \Rightarrow 2(0.75 - 0.5) = 0.75^k \Leftrightarrow 0.5 = 0.75^k \Rightarrow k = \log 0.5 / \log 0.75 \approx 2.409$
   b) $q = 1 - \varepsilon \Rightarrow (1-\varepsilon)^k = (1 - \varepsilon - \frac{1}{2})/\frac{1}{2} \Rightarrow 1 - k\varepsilon + \mathcal{O}(\varepsilon^2) = 1 - 2\varepsilon \Rightarrow k = 2 + \mathcal{O}(\varepsilon)$
   c) Сделать $a + b$: $a \in \mathbb{Z}, b \in [0, 1)$ вызовов $\Leftrightarrow$ сделать a вызовов и $(a+1)$-й с вероятностью b.

10. **Оптимизированный поток**

    Мы посетим всего V вершин. *Задача:* научиться ходить в непосещённую вершину за $\mathcal{O}(V/w)$.
    Пусть `used` – `bitset` посещённых вершин $\Rightarrow$
    Когда стоим в вершине `v`, переходим в младший бит `g[v] & ~used`.

11. **(*) Малхотра-Кумар-Махешвари**

    Оставим в графе только вершины v: $\exists s \rightsquigarrow v \rightsquigarrow t$. На оставшемся графе определим:

    $$c[v] = \min(in[v], out[v]),\ in[v] = \sum_{e \in in[v]} c_e, out[v] = \sum_{e \in out[v]} c_e$$

    Выберем за $\mathcal{O}(V)$ среди v: $c[v] > 0$ вершину x: $c[x] = \min$. Заметим, что из v в t поток можно толкать жадно – ему всегда есть куда утечь. Аналогично из s в v (толкаем из v по обратным рёбрам). Время на проталкивания $c[v]$ единиц потока из v в s и t равно $V + k_i$, где k_i – количество рёбер, по которым произошло насыщающее проталкивание. Если ребро насытилось, его сразу можно удалить из графа $\Rightarrow \sum k_i \leqslant E \Rightarrow$ суммарное время работы алгоритма $\mathcal{O}(V^2 + E)$.

Домашнее задание

3.1. Обязательная часть

1. **(1.5) Глобальный вершинный разрез**

   Удалить в связном неориентированном графе минимальное число **вершин** так, чтобы граф потерял связность. $\mathcal{O}(\texttt{Polynom}(V, E))$.

   Внимание: решение «перебрать одну вершину не работает». Почему?

   (+0.5) Оценить время работы решения, использующего Диница.

2. **(3) Округление матрицы**

   Дана матрица из вещественных положительных чисел. Необходимо так округлить вверх или вниз до целых все элементы матрицы, чтобы суммы в строках и столбцах тоже округлились вверх или вниз до целых: даны $a_{ij} \in \mathbb{R}$, найти $b_{ij} \in \mathbb{Z}$:
   $b_{ij} = \lceil a_{ij} \rceil \vee b_{ij} = \lfloor a_{ij} \rfloor$ и $\forall_i \sum_j b_{ij} = \lceil \sum_j a_{ij} \rceil \vee \sum_j b_{ij} = \lfloor \sum_j a_{ij} \rfloor$, то же самое для $\forall$ столбца j.

3. **(2) Улучшаем Каргера-Штейна**

   Модифицируйте алгоритм для работы с произвольным ($\mathbb{R}^+$) пропускными способностями. Обоснуйте время работы $\mathcal{O}(V^4)$ простой версии алгоритма.

   (a) **(1)** $c_e \in \mathbb{Z}$, (b) **(1)** $c_e \in \mathbb{R}$.

4. **(4) Модифицируем Каргера-Штейна**

   Пусть мы будем делать не два, а четыре рекурсивных вызова. В какой момент оптимальнее всего сделать ветвление, какая асимптотика получится? Если вы умеете давать только оценки снизу или сверху на вероятность или время работы, сделайте это.

5. **(2) Улучшаем 1-ю теорему Карзанова**

   Пусть $C' = \sum_v c[v]$, где сумма берётся по всем $c[v]$ кроме $\mathcal{O}(1)$ максимальных. Докажите, что число фаз алгоритма Диница $\mathcal{O}(\sqrt{C'})$.

   Подсказка: вспомните, как доказывать обычную версию теоремы Карзанова?

3.2. Дополнительная часть

1. **(3) Поток в планарном графе**

   Дана укладка планарного графа. Вершинам сопоставлены точки на плоскости, рёбра – отрезки между вершинами, рёбра не пересекаются. У рёбер есть пропускные способности. Граф неориентированный. Даны две вершины s и t, лежащие на одной грани. Задача: за $\mathcal{O}(Dijkstra)$ найти величину максимального потока из s в t.

2. **(4) Задача на строки?!**

   Решить за полином. Даны две строки s и p из символов «`0`», «`1`», «`?`». Нужно заменить все «`?`» на нули и единицы так, чтобы $d(s, p)$ было минимально.

   Здесь $d(s, p)$ равно сумме расстояний Хэмминга от p до всех подстрок s длины $|p|$.