

Второй курс, осенний семестр 2021/22

Практика по алгоритмам #5

Потоки и паросочетания

7 октября

Собрано 7 октября 2022 г. в 00:06

Содержание

1. Потоки и паросочетания	1
2. Разбор задач практики	3
3. Домашнее задание	6
3.1. Обязательная часть	6
3.2. Дополнительная часть	7

Потоки и паросочетания

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. Мальчики, девочки, собачки

Вспомним 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 $\leq a_i$.

За сколько будет работать алгоритм Диница в данном случае?

4. Татт, Ловас, Рабин и их друзья

Дан произвольный граф. Найдите размер максимального паросочетания.

Найдите само максимальное паросочетание.

5. Взвешенное покрытие циклами

Дан взвешенный орграф. Покрыть все его вершины простыми непересекающимися циклами минимального суммарного веса.

6. Покрытие минимальным числом циклов

За сколько вы можете покрыть все вершины орграфа минимальным числом простых циклов?

7. Теорема Дилворта

Есть множество, на его элементах задан частичный порядок, описанный как некий орграф.

Найти и предъявить максимальное по размеру множество попарно несравнимых элементов.

8. 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 ($l_e \leq f_e \leq c_e$).

e) Найти любой LR-поток из s в t ($l_e \leq f_e \leq c_e$).

f) Найти максимальный LR-поток из s в t ($l_e \leq f_e \leq c_e$).

9. Стабильные паросочетания

a) Приведите пример, когда стабильное паросочетание не единственно.

b) Найдите стабильное паросочетание, которое максимизирует \sum профит для мальчиков.

c) Проверить единственность стабильного паросочетания.

10. Взвешенное удаление графа

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

11. Оптимизируем КарШтейна

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

12. (*) Оптимизированный поток

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

13. (*) Варьируем константы КарШтейна

- 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?

14. (*) Покрытие циклами неорграфа

Дан произвольный неориентированный граф.

Покрыть все вершины простыми непересекающимися циклами.

15. (*) Быстрая рёберная раскраска

Пусть d – максимальная степень вершины. Идея алгоритма раскраски рёбер графа в d цветов: добавляем рёбра по очереди, у концов есть свободные цвета a и b ...

Задача: придумайте алгоритм покраски рёбер двудольного графа в d цветов за $\mathcal{O}(VE)$. Почему так можно только для двудольного?

16. (*) Малхотра-Кумар-Махешвари

Идея для оптимизации Диница: давайте научимся за $\approx \mathcal{O}(V)$ насыщать самую «узкую» вершину. 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. Мальчики, девочки, собачки

$c_v \leq \deg_v \Rightarrow \sum c_v \leq 2E \Rightarrow$ 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. Татт, Ловас, Рабин и их друзья

Решение без линейной алгебры.

Бинпоиском ищем $\max |M| = k$, внутри бинпоиска добавляем $n - 2k$ фиктивных вершин, соединённых со всеми, проверяем наличие совершенного паросочетания.

Осталось восстановить совершенное паросочетание.

Перебираем p_1 – с кем соединена первая вершина, удаляем 1 и p_1 из графа, проверяем наличие совершенного (альтернатива: перебирать ребро, удалять ребро). В обоих случаях нам нужно будет E раз считать определитель. Итого: $\mathcal{O}(V^3E)$.

Можно заметить, что при переборе p_1 матрицы, от которых мы считаем определители, отличаются лишь одной строкой, поэтому за $\mathcal{O}(V^3)$ можно найти их всех \Rightarrow суммарное время работы $\mathcal{O}(V^4)$.

5. Взвешенное покрытие циклами

Так же, как невзвешенное, раздвоим граф и ищем совершенный парсоч. Каждое ребро парсоча войдет в покрытие, значит, надо парсоч минимального веса. $\mathcal{O}(n^3)$.

6. Покрытие минимальным числом циклов

Если есть Гамильтонов цикл, то ответ – это ровно он. Значит, задача не проще Гамильтонова цикла. Динамикой по подмножествам за $\mathcal{O}^*(2^n)$ она решается.

7. Теорема Дилворта

Алгоритм. Транзитивно замкнем граф. Раздваиваем вершины v на v_L и v_R , проводим рёбра исходного графа из левой доли в правую. В полученном графе находим min вершинное покрытие, выкидываем из исходного графа вершину v , если хоть одна из v_L или v_R вошла.

Корректность. Для любого ребра (v, u) хоть одна из v_L, u_R вошла в покрытие, значит, выкинем хоть одну из v или u , значит, в исходном графе осталось независимое множество вершин («антицепь»).

Оптимальность. \max антицепь $\leq \min$ число путей, не более элемента на каждом пути. Если найти ту, которая \geq числа путей, то победа.

Если размер паросочетания k , то путей $n - k$. Размер покрытия тоже k , после выкидывания осталось $\geq n - k$ вершин исходного графа.

Теорема Дилворта как раз и гласит, что максимальный размер антицепи равен минимальному размеру разбиения на пути.

8. 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 ($l_e \leq f_e \leq c_e$).**

1. \forall ребра e пустим l_e единиц потока.

Для ребра $e: a \rightarrow b$ 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 ($l_e \leq f_e \leq c_e$).**

Добавим ещё и ребро $t \rightarrow s$ пропускной способности $+\infty$.

f) **Найти максимальный LR-поток из s в t ($l_e \leq f_e \leq c_e$).**

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 .

9. Стабильные паросочетания

a) Пример, когда не единственное: списки мальчиков $A_1 = \{1, 2\}, A_2 = \{2, 1\}$, а списки девочек наоборот $B_1 = \{2, 1\}, B_2 = \{1, 2\}$. Оба паросочетания стабильны.

b) Алгоритм «мальчики предлагает, девочки отказывают» выкидывает только те рёбра, которые не входят ни в одно стабильное паросочетание.

Действительно: пусть $A_i.\text{front} = A_j.\text{front} = k$ и девочка k отказала мальчику j .

Рассмотрим любое паросочетание, где они образуют пару: k больше хочет к i , а i больше хочет к k , т.к. она для него самая лучшая \Rightarrow паросочетание не стабильно.

В итоге каждый мальчик будет в паре с лучшей для себя из гипотетически возможных девочек. То есть оптимизирован даже каждый отдельный мальчик.

c) Алгоритм «девочки предлагают, мальчики отказываются» даёт оптимальный для каждой девочки ответ.

Запустим оба алгоритма. Если ответы не совпали, то не единственное. Если совпали, то любое другое паросочетание не стабильно: если в паросочетании нет какой-то пары из нашего, то добавим ее, этой паре стало лучше.

10. 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.

11. Оптимизируем КарШтейна

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

12. (*) **Оптимизированный поток**

Мы посетим всего V вершин. *Задача:* научиться ходить в непосещённую вершину за $\mathcal{O}(V/w)$.
`unused – bitset` непосещённых вершин.

Стоим в вершине v , переходим в младший бит `g[v] & unused`.

13. **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 .

14. (*) **Покрытие циклами неорграфа**

Пусть у вершины v степень d . Заведём вершины $v_{L,1}, \dots, v_{L,d}, v_{R,1}, v_{R,(d-2)}$, строим полный двудольный граф на наборах v_L, v_R .

Пусть соседи каждой вершины как-то упорядочены. Если есть ребро (v, u) , где v – i -й сосед u , а u – j -й сосед v , то проводим ребро $(v_{L,j}, u_{L,i})$. В полученном графе ищем совершенное паросочетание.

В совершенном покрыты все v_R , значит, из левой половины v во внешний мир выходит ровно два ребра. Получилось как раз покрытие циклами.

Если изначально двудольный гаджет каждой вершины инициализировать паросочетанием размера $(d - 2)$, то останется набрать $2V$ ребер, $\mathcal{O}(V)$ запусков поиска дополняющего пути. Ребер в новом графе $\mathcal{O}(E + \sum \deg_v^2)$. Но можно в каждой двудольном гаджете провести Cd ребер так, чтобы для \forall подмножества v_L существовало паросочетание, покрывающее его. Здесь C – константа. Например, если из каждой вершины левой доли провести 4 случайных ребра в правую, с высокой вероятностью нам повезет. Так можно сделать $\mathcal{O}(E + \sum \deg_v) = \mathcal{O}(E)$ ребер.

Итого можно решить задачу за $\mathcal{O}(\text{Matching}(V, E))$.

15. (*) **Быстрая рёберная раскраска**

Хотим добавить ребро (v, u) , $\exists a, b$: у вершины v нет ещё цвета a , у u нет b . Если $a = b$, красим, всё ок. Иначе пытаемся сделать, чтобы у u тоже не было a , берём соседнее ребро цвета a и перекрашиваем в b , а дальше перекрашиваем $b \rightarrow a$ и т.д. пока не упрёмся. Если умеем делать переход вперёд за $\mathcal{O}(1)$, всё перекрашивание займёт $\mathcal{O}(V)$.

16. (*) **Малхотра-Кумар-Махешвари**

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

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

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 $\Rightarrow \sum k_i \leq E \Rightarrow$ суммарное время работы алгоритма $\mathcal{O}(V^2 + E)$.

Домашнее задание

3.1. Обязательная часть

1. (1.5) Самолёты

Есть m самолётов, у каждого самолёта есть время вылета t_i и вместимость $1 \leq k_i \leq 10$. Есть n пассажиров, у каждого есть допустимый отрезок времён вылета $[l_i, r_i]$. Все самолёты и пассажиры летят/хотят из одного и того же пункта A в один и тот же пункт B . Нужно отправить максимальное число пассажиров.

2. (3) Подпоследовательности и gcd

Разбить массив на подпоследовательности так, что сумма $\gcd(a, b)$ по всем парам $\langle a, b \rangle$ соседних элементов всех последовательностей максимальна. $\mathcal{O}(n^3)$. Каждый элемент массива должен попасть ровно в одну подпоследовательность.

3. (3) ATSP

Дан полный граф. Граф задан несимметричной матрицей неотрицательных весов. Задача $\max\text{ATSP}$ — найти в нём максимальный по весу гамильтонов цикл. Веса неотрицательны, и удовлетворяют неравенству треугольника. Найдите 2-ОПТ приближение для $\max\text{ATSP}$ за полиномиальное время.

Подсказка: многими циклами мы уже покрывали, да? а как сделать из них один, ухудшив не больше чем в два раза?

4. (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 .

5. (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) $c_e \in \mathbb{R}$.

6. (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. (2) Школьное расписание

В школе на новый учебный год пытаются построить расписание уроков на понедельник. Уже известно, какие будут уроки, для каждого урока известно, какой учитель его ведёт, в каком кабинете урок должен проходить. Учитель может вести несколько уроков, но обязательно в разные моменты времени. В кабинете могут проходить разные уроки, но обязательно в разные моменты времени. Все уроки имеют одинаковую длину 1 час.

Учёба начинается в 8:00. Вам нужно для каждого урока выбрать время так, чтобы момент, когда все уроки закончились был как можно раньше.

2. (3) Тяжёлые самолёты

Решите задачу про самолёты для случая 10^6 самолётов, 10^6 пассажиров, вместимость каждого самолёта до 10^6 , среди пассажиров есть те, кого обязательно отправить (например, студенты, участники олимпийских игр).

3. (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 .

4. (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 $|p|$.