

2 курс ПМИ, осень 2024/25
Практика по алгоритмам #5

mincost потоки

7 октября

Собрано 7 октября 2024 г. в 13:58

Содержание

1. mincost потоки	1
2. Разбор задач практики	4
3. Домашнее задание	8
3.1. Обязательная часть	8
3.2. Дополнительная часть	9

mincost потоки

1. Mincost flow

Найдите mincost поток в орграфе. Размер потока не важен.

- Сведите задачу к mincost circulation.
- Сведите задачу к mincost k-flow. Бинпоиском и без бинпоиска.
- Подумайте, отличается ли $c_e, w_e \in \mathbb{Z}$ от $c_e, w_e \in \mathbb{R}$.

2. LR через mincost

Найти LR-поток, используя mincost flow, за время $\mathcal{O}(\text{mincost flow})$.

3. Подпоследовательности

Дан массив целых чисел, выбрать в нём k возрастающих непересекающихся подпоследовательностей. максимальной суммарной длины.

- $k = 1$. Простое решение за $\mathcal{O}(n \log n)$.
- $k = 2$. Простое решение.
- $k = 2$. Решение потоками.
- $k = 2$. Решение потоками за $\mathcal{O}(n^2)$.
- $\forall k$ решение за $\mathcal{O}(kn^2)$.
- Обобщение: дан ациклический орграф, у каждой вершины есть вес, выбрать k вершинно непересекающихся простых путей так, чтобы сумма весов выбранных вершин была максимальна. $\mathcal{O}(E + kV^2)$.
- Что изменится в пункте (f) для графа с циклами? Подумайте про $k = 1$. Что не так в рассуждении «mincost же работает, наш ответ корректен»?

4. Неправильный поток

Даны числа f_e и c_e , найдите поток $\langle f^*, c^* \rangle$ из s в t такой, что

- $\sum_e (|f_e^* - f_e|) \rightarrow \min, \quad \forall e \ c_e^* = +\infty$
- $\sum_e (|f_e^* - f_e|) \rightarrow \min, \quad \forall e \ c_e^* = c_e$
- (*) $\sum_e (|f_e^* - f_e| + |c_e^* - c_e|) \rightarrow \min$

5. Транспортная задача

В городе есть дороги, заводы-производители и магазины-дистрибьюторы. Дороги образуют орграф, у каждой дороги есть длина w_i и пропускная способность u_i . Каждый день i -й завод выпускает A_i единиц товара, j -й магазин продаёт B_j единиц товара. Составьте план доставки товара от заводов к магазинам так, чтобы $\sum f_i w_i$ пройденных дорог была минимальна.

Рассмотрим жизненную версию: $u_i = +\infty$ (по дороге можно пустить сколько угодно грузовиков, стоимость потока = стоимость бензина на перевозку). Как проще можно решить такую задачу? Как думаете, есть ли жадное решение?

6. Mincost + Диниц

Помните, как Эдмондс-Карп улучшился до Диница? Помните, что mincost k -flow – почти Эдмондс-Карп, но Форд-Беллман вместо bfs? Попробуем улучшить алгоритм для mincost.

Рассмотрим такой алгоритм для поиска mincost потока: пока \exists дополняющий путь в G_f , выделяем сеть кратчайших путей в G_f , ищем максимальный поток в этой сети.

- Докажите корректность алгоритма.
- За сколько работает такой алгоритм?
- К каким графам его разумно применять?

7. Задача про автоматы

Есть k автоматов и n заданий. Про каждое задание известно, во сколько его нужно начать делать, во сколько закончить, а также его стоимость. Каждый автомат может выполнять только одно задание в каждый момент времени. Нужно выполнить задания максимальной суммарной стоимости.

- Решение за $\mathcal{O}(kn^2)$.
- Решение за $\mathcal{O}(kn \log n)$.
- (*) Уменьшите число вершин в два раза.

8. Жадная неотрицательность

Пусть в графе есть только (а) рёбра: $w_e \geq 1$ и рёбра: $w_e = -1$.

Все отрицательные рёбра исходят из одной вершины v . Какие потенциалы применить, чтобы отрицательных рёбер не стало? (b) (*) если есть $w_e = 0$?

9. Вершинно-взвешенное паросочетание

Дан двудольный граф. У вершин есть неотрицательные веса. Вес ребра равен сумме весов его концов. Найти паросочетание максимального веса.

- $\mathcal{O}(V^3)$.
- $\mathcal{O}(VE)$. dfs от алгоритма Куна.

10. Min mean cycle canceling

$c_e \in \mathbb{R}^+$, $w_e \in \mathbb{R}$.

Если в остаточной сети нет отрицательных рёбер, mincost circulation можно найти за $\mathcal{O}(1)$.

Мы можем менять веса рёбер в остаточной сети двумя способами:

применить потенциалы, пустить по циклу поток.

Обозначим w_e^p – вес ребра с учётом потенциала p , G^p – граф с пересчитанными весами.

- Какие потенциалы p нужно применить, чтобы $\min_e w_e^p \rightarrow \max$ (обозначим $\max = x$)?
- Найдём min mean cycle, толкнём по нему поток. Что произойдёт с x ?
- (*) Посмотрите на G^p : $w_e^p \geq 0$. Через сколько итераций x должен увеличиться?
- (*) Посмотрите на G с исходными весами. Через сколько итераций min mean cycle пройдёт по $w_e \geq 0$? Докажите, что через t итераций x увеличится хотя бы в $1 - \frac{1}{n}$ раз.
- (*) Оцените время поиска mincost циркуляции с помощью ММСС.

11. (*) Непрерывная цена, мультипаросочетание

Дан двудольный граф. Нужно найти мультипаросочетание – выбрать подмножество рёбер, у каждой вершины ограничена степень. Запишем задачу формально.

$f_{ij} \in \mathbb{Z}$ – поток между i -й вершиной первой доли и j -й вершиной второй доли.

$$0 \leq f_{ij} \leq 1, s_i = \sum_j f_{ij} \leq a_i, t_j = \sum_i f_{ij} \leq b_j.$$

Пусть стоимость мультипаросочетания равна $\sum_i cost_i \cdot s_i^2$ (в обычном mincost потоке $\sum_i cost_i \cdot s_i$). Минимизируйте стоимость максимального по $\sum_i s_i$ (размеру) паросочетания.

12. (*) Регионы памяти. Расписание выполнения программ.

а) Есть k регионов памяти и n программ. У каждого региона есть размер s_i . У каждой программы есть необходимое ей количество памяти x_j и время выполнения t_j . Каждой программе нужно сопоставить номер региона памяти i_j , в котором она будет выполняться, и отрезок времени выполнения $[l_j, l_j + t_j = f_j)$. Для каждого региона памяти отрезки времени выполнения программ не должны пересекаться. Минимизировать $\sum_j f_j$.

б) Усложним задачу. Теперь у каждой программы есть вектор пар $\langle x_j, t_j \rangle$ – если предоставить программе хотя бы x_j памяти, она будет выполняться t_j времени.

13. (*) Кредитные операции - 2 (повторим)

По заданной матрице a_{ij} найти такие вектора x и y , что $x_i + y_j \geq a_{ij}$, а $\sum x_i + \sum y_j \rightarrow \min$. Дополнительно известно, что матрица или квадратная, или неотрицательная.

14. (**) Котики должны спать и есть.

Дано $n \leq 1000$ минут. Каждую минуту кот или ест, или спит (всю минуту).

В каждые k подряд идущих минут кот должен есть хотя бы e минут, спать хотя бы s минут.

Даны числа $x_i, y_i \geq 0$. Постройте расписание для кота, максимизирующее $\sum_{i \in S} x_i + \sum_{i \in E} y_i$, где S – множество минут, когда кот спит, а E – множество минут, когда кот ест.

Разбор задач практики

1. Mincost flow

Сведение к mincost circulation: добавить ребро $e: t \rightarrow s$, $c_e = +\infty$, $w_e = 0$.

Сведение к mincost k-flow. Пусть f_k – оптимальный поток размера k , $path_k$ – кратчайший путь в остаточной сети G_{f_k} . Вспомним, что $W(path_k) \nearrow \Rightarrow$ найдём $\min k: W(path_k) \geq 0$.

Можно искать k бинпоиском, можно линейным (толкая $\min_e(c_e - f_e)$ по очередному пути).

Линейный поиск даст время $\mathcal{O}(VE \cdot \text{Dijkstra})$, также как и простейший mincost k-flow.

Внутри бинпоиска можно использовать более крутые решения. Например, сперва Диниц с link-cut ищет любой поток h размера k , затем capacity scaling ищет mincost circulation в $G_h \Rightarrow \mathcal{O}(VE \log V + \text{Dijkstra} \cdot E \log U)$.

Если $c_e \in \mathbb{R}$, то log бинпоиска и log от capacity scaling становятся вида $\log \frac{U}{\epsilon}$.

2. LR через mincost

Вместо LR-ребра делаем L -ребро стоимости $-\infty$ и $(R-L)$ -ребро стоимости 0 . Если в полученном графе есть отрицательные циклы – ничего страшного, задачу mincost circulation мы тоже умеем решать.

3. Подпоследовательности

Решение динамикой для $k = 1$. minEnd[length] за $\mathcal{O}(n \log n)$.

Решения динамикой для $k = 2$.

sumLength[end1, end2] за $\mathcal{O}(n^3)$, minEnd2[end1, sumLength] за $\mathcal{O}(n^2 \log n)$.

Mincost flow.

Раздваиваем вершины, $e: i \rightarrow j, i < j, a_i < a_j, w_e = -1, c_e = 1$. Время работы $k \cdot VE = k \cdot n^3$.

Дейкстра. Исходный граф ациклический \Rightarrow можно за $\mathcal{O}(V + E)$ динамикой (dfs) найти кратчайшие пути, использовать их, как потенциалы, далее пускать Дейкстру. $\mathcal{O}(kn^2)$.

Ациклический граф. Раздваиваем вершины, помним про dfs для поиска кратчайших путей, пускаем поток Дейкстрой.

Что не так с циклами? Для $k = 1$ нас просят найти гамильтонов путь.

4. Неправильный поток

a) $\sum_e (|f_e^* - f_e|) \rightarrow \min, \forall e c_e^* = +\infty$.

Сделаем две копии ребра: f_e потока стоимостью -1 , и $+\infty$ стоимостью 1 .

Другое решение. Пусть f_e потока жадно по каждому ребру. Образовались избытки и недостатки. Нужно избытки перенаправить в недостатки. Чтобы перенаправить единицу потока, придётся поменять f на каждом ребре пути \Rightarrow стоимость изменения равна длине пути \Rightarrow нужно минимизировать $\sum_e \Delta f_e w_e$ для $w_e = 1$. Это задача mincost flow.

b) $\sum_e (|f_e^* - f_e|) \rightarrow \min, \forall e c_e^* = c_e$. То же самое, но нельзя увеличивать больше чем до $c_e \Rightarrow$ пропускная способность прямого ребра $c_e - f_e$, а обратного f_e .

c) $\sum_e (|f_e^* - f_e| + |c_e^* - c_e|) \rightarrow \min$. То же самое, но чтобы увеличивать больше чем c_e нужно платить $2 \Rightarrow$ прямое ребро состоит из двух частей: $\langle c_e - f_e, 1 \rangle, \langle +\infty, 2 \rangle$.

5. Транспортная задача

Как обычно, перенаправим избытки (A_i у заводов) в недостатки (B_j у магазинов). Для этого добавляем исток, сток и ищем mincost max flow.

Если $\forall e u_e = +\infty$, можно найти матрицу расстояний, составить двудольный граф. Если $A_i = B_j = 1$, получаем задачу о назначениях \Rightarrow mincost flow или венгерка, но не жадность.

6. Mincost + Диниц

Пусть перед очередной фазой наш поток f , G_f – остаточная сеть, $H(G_f)$ – сеть кратчайших путей. $d(s, t)$ – длина кратчайшего пути от s до t в G_f .

Корректность.

\forall путь из s в t в $H(G_f)$ – кратчайший из s в t в G_f . Мы уже знаем, что $d(s, t) \nearrow$.

Пусть $\Delta f = \max$ поток в $H(G_f)$, декомпозируем его на пути, все они веса d , циркуляция в декомпозиции неотрицательна $\Rightarrow W(f + \Delta f)$ оптимален.

Время работы.

В отличие от Диница сеть кратчайших путей $H(G_f)$ – не слоистый граф, а произвольный. Например, пусть $\forall e w_e = 0 \Rightarrow H = G_f$ (сеть кратчайших путей содержит все рёбра G_f) \Rightarrow нельзя пустить одну фазу Диница, нужно пускать полноценный поток.

После одной фазы «пустим Диница» длина кратчайшего пути возрастёт (от противного): $G_{f+\Delta f}$ содержит путь p длины $d \Rightarrow$ рассмотрим поток $(\Delta f) + p$ в G_f , декомпозируем его на пути, все пути имеют длину ровно $d \Rightarrow (\Delta f) + p$ – поток в $H(G_d) \Rightarrow \Delta f$ не максимален. \Rightarrow время работы $\mathcal{O}(D(V E + \max \text{Flow}))$, где D – число различных кратчайших расстояний.

Работает быстро, если D мало.

Если все цены рёбер в графе равны, различных расстояний не более V , получается $\mathcal{O}(V(V E + \max \text{Flow}))$. Если все цены маленькие, $\mathcal{O}(1)$, выйдет такая же асимптотика.

7. Задача про автоматы

Можно решать так же, как предыдущую: ребро между заданиями, если одно можно начать после конца другого. Получится граф из $\Theta(n^2)$ ребер.

Быстрое решение. Вершины графа – моменты времени: начала и концы заданий. Ребро цены $-cost_i$ и пропускной способности 1 между началом и концом i -го задания. Также ребро из каждого момента времени в следующий момент, цена 0, пропускная способность $+\infty$. То есть, свободный автомат либо берет задание и освобождается к концу задания, либо переходит в следующий момент времени, оставаясь свободным. Получилось $2n$ вершин и $3n$ рёбер. Mincost поток размера k ищем за $\mathcal{O}(E + k \cdot Dijkstra) = \mathcal{O}(kE \log V) = \mathcal{O}(kn \log n)$.

Ещё меньше вершин. Заметим, что из второй половинки задания всегда исходит ровно одно ребро. Зачем нам такая вершина? Получаем n вершин и $2n$ рёбер.

8. Жадная неотрицательность

а) Увеличим потенциал v на 1.

б) Чтобы не появились входящие в v рёбра веса -1 , увеличим на 1 потенциал всех $u \in S$, где S – вершины, достижимые из v по входящим нулевым рёбрам.

9. Вершинно-взвешенное паросочетание

Венгерка и mincost дают $\mathcal{O}(V^3)$. Внутри mincost важно использовать Дейкстру.

Чтобы получить решение за $\mathcal{O}(VE)$ заметим, что вес дополняющего пути равен сумме весов конечных вершин. Давайте за $\mathcal{O}(E)$ найдём пару вершин i, j : $a_i + b_j = \min$ и есть путь $i \rightsquigarrow j$. Перебираем i в порядке a_i и для каждого i ищем из неё путь в dfs-ом в минимальное b_j , игнорируя те j , которые мы нашли для предыдущих i (т.е. не обнуляя пометки между dfs).

10. Min mean cycle canceling

[Lecture in english]

x = весу цикла минимального среднего веса.

Как найти потенциалы? Найти x , и в графе $w'_e = w_e - x$ (без отрицательных циклов) найти расстояния, которые и есть потенциалы.

x может только увеличиться.

В сети G^p с весами $w'_e = w_e - x$ нет отрицательных рёбер и каждый цикл будет насыщать хотя бы одно нулевое ребро \Rightarrow за m итераций нулевые рёбра кончатся.

В сети G^p с весами w_e нет рёбер меньше $x \Rightarrow$, текущий цикл минимального среднего веса имеет все рёбра равные $x \Rightarrow$ когда очередной цикл пройдёт через ребро веса $w_e > 0$, вес цикла будет хотя бы в $1 - \frac{1}{length}$ раз больше.

11. (*) Непрерывная цена, мультипаросочетание

Когда мы уже пустили f потока в i -ю вершину левой доли, мы платим $f^2 cost_i$. Пустим ещё $+\varepsilon$ потока, получим стоимость $(f + \varepsilon)^2 cost_i$, прирост равен $(2f\varepsilon + \varepsilon^2) cost_i = 2f\varepsilon cost_i + o(\dots)$. Получаем, что удельная стоимость единицы потока равна $2f \cdot cost_i$.

Можно представить, что у нас не одно ребро, а бесконечно много узких рёбер. j -е ребра ребра в i -ю вершину левой доли стоит $2(\varepsilon \cdot j) \cdot cost_i$. В одной пачке рёбра упорядочены по возрастанию веса ($cost_i > 0$). Стоимость кратчайшего пути – стоимость его первого ребра. Алгоритм поиска mincost потока говорит, что корректно $+\varepsilon$ потока толкнуть по пути, который начинается с ребра минимального веса.

Решение: n раз бинпоиском находим такое максимальное x , что можно пустить по всем рёбрам поток $\frac{x}{2cost_i}$, таким образом, заплатив за каждое $\leq x$, $cost_i$ пересчитано в соответствии с уже пущенным потоком. Проверка внутри бинпоиска через max flow. После конца бинпоиска по одному из n рёбер в левую долю нельзя пустить даже $x + \varepsilon$ потока, его можно просто удалить. Всего из истока исходит n рёбер, в худшем случае они будут умирать по одному.

Time = $\mathcal{O}(n \log MAX \cdot MaxFlow)$.

12. (*) Регионы памяти. Расписание выполнения программ.

Рассмотрим i -й регион памяти. Пусть в нём выполняются программы j_m, j_{m-1}, \dots, j_1 . Тогда ответ для него равен $(t_{j_1} + \dots + t_{j_m}) + (t_{j_2} + \dots + t_{j_m}) + (t_{j_3} + \dots + t_{j_m}) + \dots = t_{j_1} \cdot 1 + t_{j_2} \cdot 2 + \dots$. Видно, что в одном регионе выгодно выполнять программы в порядке убывания t .

Построим двудольный граф, в котором левая доля – программы, правая доля – пары $\langle i, j \rangle$ обозначающие, что мы хотим выполнить работу в i -м регионе на j -й с конца позиции. Рёбра между долями проводим в соответствии с ограничениями на память. Цена ребра из p в $\langle i, j \rangle$ равна $j \cdot t_p$. $\mathcal{O}(nk)$ вершин, $\mathcal{O}(n^2k)$ ребер.

13. (*) **Кредитные операции - 2 (повторим)**

Венгерка на матрице $-A$ ищет такие x и y , что есть совершенное паросочетание по нулям, а остальные $-a_{ij} + x_i + y_j \geq 0 \Rightarrow$ выполнено $x_i + y_j \geq a_{ij}$.

При любом паросочетании $x_i + y_j \geq a_{ij} \Rightarrow \sum x_i + \sum y_j = \sum (x_i + y_{p[i]}) \geq \sum a_{i,p[i]}$ (тут нам пригодилась квадратность). У нас паросочетание по нулям $\Rightarrow -a_{i,p[i]} + x_i + y_{p[i]} = 0 \Rightarrow \sum x_i + \sum y_j = \sum a_{i,p[i]} \Rightarrow$ минимально.

14. (**) **Котики должны спать и есть.**

Задача D из полуфинала 2016-го года.

Домашнее задание

3.1. Обязательная часть

1. (2) Mincost поток нельзя искать bfs-ом

Рассмотрим задачу: найти в невзвешенном орграфе два рёберно непересекающихся пути из s в t минимальной суммарной длины. Мальчик Вася пытается решить задачу так «запустим два раза bfs». Покажите, что и если Вася подумал об обратных рёбрах, и если не подумал, его решение некорректно. По баллу за каждый из двух вариантов.

2. (2) Оптимальность mincost потока

Рассмотрим задачу mincost k-flow. $n, m \leq 10\,000$. Предложите строго полиномиальный алгоритм, который проверяет оптимальность найденного ответа.

3. (2.5) Mincost LR-flow

Предложите алгоритм для поиска LR-потока минимальной стоимости.

4. (4) Равномерное паросочетание 2

Дана матрица выполнимости – какой рабочий какие работы может выполнять. Распределим **все** работы между рабочими. Пусть i -му рабочему досталось v_i работ. Рассмотрим вектор $V = (v_1, v_2, \dots, v_m)$. Нужно так распределить работы, чтобы $|V_{opt} - V| \rightarrow \min$. $V_{opt} = (\frac{n}{m}, \frac{n}{m}, \dots, \frac{n}{m})$, вектор вещественных чисел, где n – число работ, m – число рабочих, $|X - Y|$ – евклидово расстояние между m -мерными векторами X, Y .

- (3) Свести задачу к потоку минимальной стоимости.
- (1) Свести задачу к алгоритму Куна или Куноподобному dfs-у.

5. (3) Автомойка

Есть n машин. Каждую нужно привести в приличный вид, для этого нужно обработать её в трёх мастерских в строго опеределённом порядке — сперва A , затем B , затем C .

В день каждая мастерская может принять не более одной машины.

Нужно уложиться в $m \geq n+2$ дней. A берёт в i -й день a_i за процедуру, $B \rightarrow b_i, C \rightarrow c_i$.

Даны n, m, a_i, b_i, c_i , найти расписание, минимизирующее суммарную стоимость процедур.

(3) $\mathcal{O}(\text{Polynom}(n, m))$

(+1) $\mathcal{O}(nm \log m)$

3.2. Дополнительная часть

1. (3) Быстрые последовательности

Модифицируйте граф для нахождения k непересекающихся возрастающих подпоследовательностей (см. практику) так, чтобы в нём было $\mathcal{O}(n \log n)$ рёбер.

Получится решение исходной задачи за $\mathcal{O}(kn \log^2 n)$.

2. (3) Подгон MST

Дан граф G , в нём выделено остовное дерево T . Мы можем уменьшать и увеличивать веса рёбер. Сделать T минимальным по весу остовным деревом. При этом минимизировать суммарное изменение весов рёбер.

Подсказка: остов минимальный iff вес любого ребра не из остова не меньше максимума на соответствующем пути. Мы на практиках решали задачу про $x_i + y_j \geq a_{ij}$.

3. (4) Увеличение длины пути

Дан орграф с неотрицательными весами рёбер.

Стоимость увеличения веса ребра e на $x \in \mathbb{R}$ равна $c_e \cdot x$.

Есть P денег, потратьте их с умом – максимизируйте стоимость кратчайшего пути из s в t .

4. (3.5) Крестьяне и поля

Дана матрица $n \times m$. В некоторых клетках горы, в некоторых живут крестьяне, в некоторых поля. Расстояние между клетками (x_1, y_1) и (x_2, y_2) считается без учёта гор: просто $|x_1 - x_2| + |y_1 - y_2|$. Полей не меньше, чем крестьян. Если фиксирован порядок крестьян p , то можно раздать поля следующим алгоритмом: в порядке p каждый крестьянин получает свободное поле, из таких ближайшее, из таких $x \rightarrow \min$, из таких $y \rightarrow \min$.

Задача: выбрать p , чтобы сумма расстояний от крестьян до их полей была минимальна.