

SPb HSE, ПАДИИ, 1 курс, весна 2024/25

Практика по алгоритмам #27

Mincost Потоки

24 апреля

Собрано 24 апреля 2025 г. в 13:41

Содержание

1. Mincost Потоки	1
2. Разбор задач практики	2
3. Домашнее задание	4
3.1. Обязательная часть	4
3.2. Дополнительная часть	4

Mincost Потoki

1. Mincost flow

Найдите mincost поток в орграфе. Размер потока при этом не важен.

Сведите задачу mincost circulation к задаче mincost flow.

Сведите задачу mincost k-flow к задаче mincost flow.

2. LR через mincost

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

Не забудьте подумать про отрицательные циклы.

3. Подпоследовательности

Дан массив целых чисел, выбрать в нём k возрастающих непересекающихся подпоследовательностей. максимальной суммарной длины. $\mathcal{O}(kn^3)$. (*) $\mathcal{O}(kn^2)$.

4. 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. (*) Придумайте граф, в котором $\mathcal{O}(n)$ рёбер.

5. Красно-синие вершины

Дан неорграф, какие-то вершины покрашены в красный, какие-то в синий.

(a) докрасить оставшиеся, чтобы пар соседних разноцветных было поменьше.

(b) число вершин, у которых есть сосед другого цвета, $\rightarrow \min$.

(c) у вершин есть веса $\sum w_v \rightarrow \min$.

6. Япония. Инструменты и работы.

Разбор домашки.

7. (*) Взвешенные Cover, IS

Дан двудольный граф. У каждой вершины есть неотрицательный вес. Найти вершинное покрытие минимального веса, независимое множество максимального веса.

Указание: вспомните с лекции, как потоками искать в двудольном графе matching, IS, VC.

8. (*) Неправильный поток

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

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

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

c) $\sum_e (|f_e^* - f_e| + |c_e^* - c_e|) \rightarrow \min$

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

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 $\sum_j f_j$.

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.

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

Mincost flow.

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

4. 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 $\mathcal{O}(E + k \cdot \text{Dijkstra}) = \mathcal{O}(kE \log V) = \mathcal{O}(kn \log n)$.

5. Красно-синие вершины

(a) Синие = исток, красные = сток. Ищем min разрез.

(b) Раздвоим вершины, исходные рёбра $c_e = \infty$, рёбра-вершины $c_e = 1$.

(c) Поменяем c_e рёбер-вершин на вес вершин.

6. Япония. Инструменты и работы.

Из истока рёбра в инструменты, $c_e = cost$.

Из работ рёбра в сток, $c_e = cost$.

Из заказа оррёбра во все инструменты, нужные для его выполнения, $c_e = +\infty$.

7. (*) Взвешенные Cover, IS

Построим сеть для поиска паросочетаний: добавим слева исток и справа сток. Исходные рёбра сделаем веса $+\infty$ и направим слева направо. Теперь есть биекция между (S, T) -разрезами без бесконечных рёбер и вершинными покрытиями:

VC = вершины левой доли из $S \cup$ вершины правой доли из T .

Отсюда решение: на ребре, соединяющем вершину v с истоком или стоком, пишем w_v , на рёбрах исходного графа пишем $+\infty$. Ищем $\min$ разрез. $\max$ IS – дополнение к $\min$ VC.

8. (*) 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: $\langle c_e - f_e, 1 \rangle, \langle +\infty, 2 \rangle$.

9. (*) 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.

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) Автомоёка

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.

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

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

3.2. Дополнительная часть

1. (3) 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 .