Второй курс, осенний семестр 2018/19
Практика по алгоритмам #2

Паросочетания, потоки

14 сентября

Собрано 15 сентября 2018 г. в 14:42

Содержание

Паросочетания, потоки

1. **Покрытие минимальным числом циклов**
   За сколько вы можете покрыть все вершины орграфа минимальным числом простых циклов?
2. **Теорема Дилворта**
   Есть множество, на его элементах задан частичный порядок, описанный как некий орграф. Найти и предъявить максимальное по размеру множество попарно несравнимых элементов.
3. **Венгры в прямоугольной матрице**
   Модифицируйте венгерский алгоритм для поиска по прямоугольной матрице паросочетания размера $\min(n, m)$. Оценить сложность.
4. **Венгерка и отрицательные веса**
   Работает ли венгерский алгоритм в матрицах, где могут быть отрицательные веса?
5. **Паросочетание максимального веса**
   Умеем искать паросочетание минимального веса. Как найти максимального?
6. **Кредитные операции - 2**
   Дана **квадратная** матрица A. Найти вектора x и y: $x_i + y_j \geqslant a_{ij}$, а $\sum x_i + \sum y_j \to \min$.
   (*) Пусть все $a_{ij} \geqslant 0$, найдите неотрицательные x и y.
7. **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.
8. **Форд-Фалкерсон не такой уж и медленный**
   Оцените время Форда-Фалкерсона на графе с единичными пропускными способностями.
9. **Крутая декомпозиция**
   Декомпозиция потока f – представление f в виде суммы элементарных потоков (путей) и циркуляции (потока величины 0).
   Научитесь декомпозировать поток на пути за $\mathcal{O}(E^2)$, потом за $\mathcal{O}(VE)$.
10. **Вершинно-непересекающиеся пути**
    Научитесь за $\mathcal{O}(kE)$ находить k вершинно непересекающихся путей из s в t.
    (a) в орграфе, (b) в неорграфе.
11. (*) **Покрытие циклами неорграфа**
    Покрыть все вершины произвольного **неор**графа простыми непересекающимися циклами.
12. (*) **Покраска дерева**
    Дано дерево. Вершину v можно покрасит в цвет $c \in [1, k]$ за $\texttt{cost}[v, c]$. Покрасить за минимальную цену все вершины так, чтобы расстояние между вершинами одинакового цвета было строго больше двух.
    a) $\mathcal{O}(nk^5)$.
    b) $\mathcal{O}(nk^4)$.
    c) (**) $\mathcal{O}(nk^3)$.
13. (*) **Хитрая матрица**
    Дана матрица A, придумайте матрицу $B\colon \forall i\, j\, b_{ij} \geqslant a_{ij}, \sum(b_{ij} - a_{ij}) \to \min$,
    при этом для матрицы B должно выполняться свойство: $\forall i, j \quad b_{i,j+1} + b_{i+1,j} = b_{i,j} + b_{i+1,j+1}$.

Разбор задач практики

1. **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.

2. **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 туда.
   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 мн-во вершин («антицепь»).
   Оптимальность. max антицепь $\leqslant$ min число путей, не более элемента на каждом пути. Если найти ту, которая $\geqslant$ числа путей, то победа.
   Если размер паросочетания k, то путей $n-k$. Размер покрытия тоже k, после выкидывания осталось $\geqslant n-k$ вершин исходного графа.
   Теорема Дилворта как раз и гласит, что максимальный размер антицепи равен минимальному размеру разбиения на пути.

3. **Венгры в прямоугольной матрице**
   Пусть $n \leqslant m$, тогда мы n раз ищем дополняющую цепь.
   В процессе поиска пути мы строим дерево на нулях, в него добавлений $\leqslant n$ (вершины добавляются парами: вершина из правой доли и ее пара из левой, а в левой n).
   Добавление в дерево мы можем сделать за m, поиском вершины правой доли, в которую ведет минимальное ребро.
   После добавления в дерево пересчитываем $\texttt{row}_i$ и $\texttt{col}_j$ за $n+m = \mathcal{O}(m)$.
   Итого $\mathcal{O}(n^2m)$.

   Есть загвоздка с корректностью. В квадратной матрице мы могли менять вес столбца, поскольку это меняло вес любого паросочетания на константу. Здесь это не так: некоторые столбцы мы не возьмём.
   Мысленно дополним матрицу до квадратной добавлением вниз строк, заполненных $+\infty$. Теперь всё корректно.
   Но мы никогда не смотрим в строки ниже текущей, значит, явно добавлять эти строки не надо. Т.е., обычная реализация работает на прямоугольной матрице.

4. **Венгерка и отрицательные веса**
   Венгерка строит паросочетание по нулям. Если в конце $a_{ij} + \texttt{row}_i + \texttt{col}_j \geqslant 0$, то оно правда минимально.
   Пусть пришли в строку i. Сейчас в A^+ только она, в B^- все столбцы $\Rightarrow$ первым же шагом мы просто вычтем из строки i ее минимальный элемент, и она станет неотрицательной.
   При построении дерева из строки i, мы работаем только со строками $\leqslant i$, там уже все элементы $\geqslant 0$. Значит, когда мы делаем $\texttt{+}x$ на $A^- \times B^+$, мы только увеличиваем числа $\Rightarrow$ строки $\leqslant i$ будут оставаться неотрицательными.

5. **Паросочетание максимального веса**

max паросочетание на A = min паросочетание на $-A$.

6. **Кредитные операции - 2**
Венгерка на матрице $-A$ ищет такие x и y, что есть совершенное паросочетание по нулям, а остальные $-a_{ij} + x_i + y_j \geqslant 0$.
Тогда выполнено $x_i + y_j \geqslant a_{ij}$.
При любом паросочетании $x_i + y_j \geqslant a_{ij} \Rightarrow \sum x_i + \sum y_j = \sum(x_i + y_{p[i]}) \geqslant \sum a_{i,p[i]}$ (тут нам пригодилась квадратность).
У нас паросочетание по нулям $\Rightarrow -a_{i,p[i]} + x_i + y_{p[i]} = 0 \Rightarrow \sum x_i + \sum y_j = \sum a_{i,p[i]}$. Значит, минимально.

7. **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 $A_i.\mathtt{front} = A_j.\mathtt{front} = k$ 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 отдельгный мальчик.
   c) Алгоритм «девочки предлагают, мальчики отказываются» дает 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.

8. **Форд-Фалкерсон не такой уж и медленный**
$|f| \leqslant E \Rightarrow$ время работы $\mathcal{O}(E^2)$.
Точнее $|f| \leqslant \min(\deg_s, \deg_t)$. Без кратных ребер $\mathcal{O}(VE)$.

9. **Крутая декомпозиция**
Заставим dfs работать за $\mathcal{O}(V)$. Для этого
   a) Если он по ребру не нашёл путь, удалим ребро
   b) Если он зациклился, отменим поток по циклу

   В сумме k dfs-ов отработают за $\mathcal{O}(E + kV) \Rightarrow$ декомпозиция за $\mathcal{O}(VE)$.

10. **Вершинно не пересекающиеся пути**
Орграф: раздваиваем вершину, посередине ребро пропускной способности 1, входящие рёбра в первую половину, исходящие из второй.
Неорграф: приводим к орграфу, делаем то же самое.

11. **(*) Покрытие циклами неорграфа**
Пусть у вершины v степень d. Заведем вершины $v_{L,1}, \ldots, v_{L,d}, v_{R,1}, v_{R,(d-2)}$, 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 $\mathcal{O}(\texttt{Matching}(V,E))$.

12. **(*) Покраска дерева**
   a) $\mathcal{O}(nk^5)$. Динамика $d[v,c,c_p]$ – минимальная цена покрасить поддерево с корнем v в цвет c, а цвет предка c_p. Детей красим в разные цвета, отличные от c, c_p. Венгеркой: соединяем ребенка u с цветом a ребром веса $d[u,a,c]$, ищем минимальное паросочетание детей и цветов.
   b) $\mathcal{O}(nk^4)$. При фиксированных v, c мы перебираем c_p, при этом один цвет становится недоступен, один доступен.
   Не будем строить всё паросочетание с нуля. Если ушел цвет a, то его пара u осталась без пары, поищем дополняющий путь из u за k^2.
   Итого для каждых v, c сначала за k^3 ищем ответ для $c_p = 0$, потом $k-1$ раз при смене c_p ищем дополняющий путь за k^2.
   c) $\mathcal{O}(nk^3)$. Преподавательский состав не умеет, но не теряет надежды, что студенты что-нибудь придумают.

13. **(*) Хитрая матрица**
   $b_{i,j+1} + b_{i+1,j} = b_{i,j} + b_{i+1,j+1} \Leftrightarrow b_{i,j+1} - b_{i,j} = b_{i+1,j+1} - b_{i+1,j}$.
   Отсюда видим, что каждый столбец получается из следующего прибавлением к нему одного и того же числа.
   Значит, если x – первый столбец матрицы B, то есть некий вектор y: $b_{ij} = x_i + y_j$.
   Получили задачу «Кредитные операции - 2».

Домашнее задание

3.1. Обязательная часть

1. **(2) Самолёты**
   Есть самолёты, у каждого самолёта есть время вылета t_i и вместимость $1 \leqslant k_i \leqslant 10$. Есть пассажиры, у каждого есть допустимый отрезок времён вылета $[l_i, r_i]$. Все самолёта 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) Сделаем венгерский алгоритм полезнее**
   a) **(1.5)** Найдите максимальное по весу паросочетание произвольного размера.
   b) **(1.5)** Найдите максимальное по весу паросочетание размера ровно k.
3. **(3) ATSP**
   Дан полный граф. Граф задан несимметричной матрицей неотрицательных весов. Задача maxATSP – найти в нём максимальный по весу гамильтонов цикл. Веса и неотрицательны и удовлетворяют неравенству треугольника. Найдите 2-OPT приближение для maxATSP за полиномиальное время.
4. **(2) Единственность минимального разреза**
   Дан граф и выделенные вершины s и t. За $\mathcal{O}(\texttt{Polynom}(V, E))$ проверить, правда ли, что существует **единственный** минимальный разрез?
5. **(3) Задача про трисочетание**
   Даны девочки, мальчики и собачки.
   Для каждой пары «мальчик, девочка» известно, хочет ли девочка дружить с мальчиком.
   Для каждой пары «собачка, девочка» известно, нравится ли собачка девочке.
   Нужно максимальному количеству девочек выделить по мальчику и собачке так, что:
   a) каждый мальчик не более чем с одной девочкой;
   b) каждая собачка не более чем у одной девочки;
   c) девочка хочет дружить с выбранным ей мальчиком, и собачка ей нравится.
6. **(3) Равномерное распределение**
   Есть n рабочих и m работ. И есть матрица умений: «какой рабочий какие работы умеет делать». Нужно максимально равномерно распределить работы между рабочими. То есть, каждой работе сопоставить рабочего, который умеет делать эту работу, а кроме того минимизировать $\max\limits_{i=1..n} k_i$, где k_i – количество работ, выданных i-му рабочему.

3.2. Дополнительная часть

1. **(1.5) Татт, Ловас, Рабин и их друзья**
   Предложите полиномиальный (и, возможно, вероятностный) алгоритм поиска паросочетания в произвольном графе.
2. **(2.5) Уравнивание деревьев**
   Даны два корневых дерева. У каждого дерева можно отрезать вершины с их поддеревьями, менять порядок детей у вершины. Сделать данные деревья равными, максимизировав число оставшихся вершин.
3. **(3) Тяжёлые самолёты**
   Решите задачу про самолёты для случая 10^5 самолётов, 10^5 пассажиров, вместимость каждого самолёта до 10^5, среди пассажиров есть те, кого обязательно отправить (например, студенты, участники олимпийских игр).
4. **(2) Единственность разреза. $\mathcal{O}(E)$.**
   Пусть дан какой-то максимальный поток.
   a) За $\mathcal{O}(E)$ проверить единственность минимального разреза. С доказательством.
   b) За $\mathcal{O}(E)$ найти минимальный разрез $V = S \sqcup T\colon |S| = \max$.
5. **(3) Тест против Форда-Фалкерсона**
   Найдите ориентированный граф с целочисленными пропускными способностями, на которых детерминированный алгоритм Форд-Фалкерсона с фиксированным порядком перебора рёбер, пропускающий $\min_e(c_e - f_e)$ по пути, работает за экспоненту от V.
   (+2) балла за тест против ФФ, в котором перед каждым `dfs` делается `random_shuffle` рёбер.