Второй курс, осенний семестр 2018/19
Практика по алгоритмам #1

Паросочетания

7 сентября

Собрано 8 сентября 2018 г. в 15:34

Содержание

Паросочетания

1. **Покрытие доминошками**
Покрыть клетчатое поле $n \times m$ с дырками доминошками. Каждая клетка должна быть покрыта ровно один раз. $\mathcal{O}(\text{poly}(n, m))$.

2. **Двудольная клика**
Найти максимальную двудольную клику (полный двудольный подграф) в двудольном графе за $\mathcal{O}(VE)$.

3. **Максимизация $|A| - |N(A)|$**
Выбрать в двудольном графе множество A из левой доли, что $|A| - |N(A)| \to \max$.
$N(A)$ – множество соседей A в правой доле). $\mathcal{O}(VE)$.

4. **Удаление графа**
Дан орграф. За один ход можно удалить все входящие или все исходящие ребра одной вершины (но не одновременно). Удалить все рёбра графа за min число ходов.

5. **Покраска матрицы отрезками**
За один ход можно покрасить один произвольный вертикальный или горизонтальный отрезок матрицы в белый цвет, мазки могут перекрываться. За минимальное число ходов привести чёрную матрицу к заданному виду.

6. **Можно снимать часть пометок**
Докажите, что можно при успешном нахождении дополняющего пути снимать пометки только с вершин, посещенных в этом запуске `dfs`.

7. **Жадная инициализация не нужна**
Докажите, что первый проход «Куна» с оптимизацией «вообще не обнуляем пометки» найдет паросочетание не меньше, чем жадная инициализация ($\geqslant |M|/2$).

8. **Единственность паросочетания**
По данному максимальному паросочетанию в **двудольном** графе проверить, является ли оно единственным. $\mathcal{O}(E)$.
`(*)` А если граф недвудольный?

9. **Разбиение на плавные подпоследовательности**
Разбить массив на минимальное число подпоследовательностей таких, что в каждой подпоследовательности разность соседних элементов по модулю не превышает d. $\mathcal{O}(n^3)$.

10. **Ездят такси, но нам нечем платить**
По взвешенному графу ездят такси. Заказ на такси задается начальной вершиной, конечной вершиной и временем отправления из начальной вершины. Выполнить все заказы, используя минимальное число машин.

11. **Покрытие множества паросочетанием**
Придумайте алгоритм, который строит паросочетание, покрывающее множество A вершин первой доли за время $\mathcal{O}(VE)$, докажите его корректность.
Как найти максимальное среди всех паросочетаний, покрывающих A?

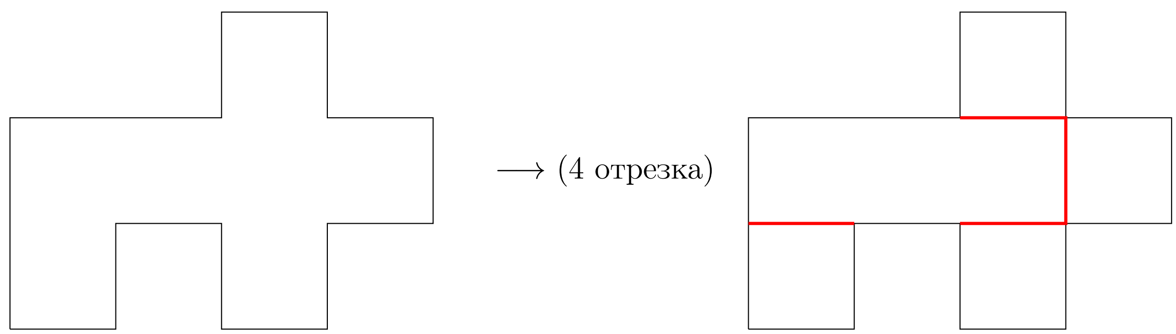
12. **Реберное покрытие**
Реберное покрытие – множество ребер, среди концов которых есть все вершины.
Найти минимальное реберное покрытие двудольного графа.

13. **Эйлеровость и паросочетания**
Научитесь красить рёбра d-регулярного двудольного графа в d цветов за $\mathcal{O}(\texttt{Matching} \cdot \log d)$.

14. **(*) Birkhoff–von Neumann decomposition**
Дана матрица из неотрицательных вещественных чисел. Сумма элементов в каждой строке и каждом столбце равна 1. Разложить заданную матрицу на сумму перестановочных матриц с коэффициентами.
a) $\mathcal{O}(n^5)$
b) $\mathcal{O}(n^4)$

Перестановочная матрица – перестановка строк единичной матрицы (т.е. в каждой строке и каждом столбце по одной единице).

15. **(*) Прямоугольные многоугольники**
Дан простой многоугольник со сторонами параллельными осям координат. Все углы прямые. Разрезать его на минимальное число прямоугольников. $N \leqslant 100$.



Разбор задач практики

1. **Покрытие доминошками**
   Невырезанной клетка – вершина графа, граница между соседними клетками – ребро. Покрытие доминошками – в точности паросочетание.
   Осталось разбить на две доли. Это можно сделать в шахматном порядке: в одной доле клетки с четной суммой координат, в другой с нечетной.

2. **Двудольная клика**
   Инвертируем ребра и находим IS.
   То есть проведем между долями все ребра, которых не было, и удалим все, которые были. Любая клика в исходном графе – независимое множество в новом.
   Время работы выйдет $\mathcal{O}(V(V^2 - E)) = \mathcal{O}(V^3)$.

3. **Максимизация** $|A| - |N(A)|$
   Пусть R – правая доля.
   Вместо $|A| - |N(A)|$ максимизируем отличающееся от него на константу $|A| - |N(A)| + |R| = |A| + |R \setminus N(A)|$.
   Это размер max независимого множества. A – пересечение IS с L.

4. **Удаление графа**
   Строим новый двудольный граф: вершине v сопоставим пару вершин v_{out} и v_{in}, ребру (v, u) – ребро (v_{out}, u_{in}).
   v_{in} покрывает ребра, входящие в v, а v_{out} – исходящие. Видно, что ответ – min VC в новом графе.

5. **Покраска матрицы отрезками**
   Рассмотрим множества H и V всех максимальных по включению горизонтальных и вертикальных белых отрезков.
   Построим двудольный граф с долями H и V. Каждой белой клетке сопоставим ребро, соединяющее два отрезка, пересекающихся в этой клетке.
   Ответ – min VC этого графа.

6. **Можно снимать часть пометок**
   Доказывая корректность Куна, мы доказали, что если из вершины нет дополняющего пути, то и никогда больше не будет.
   Значит, если мы посетили какие-то вершины в неуспешном запуске, то из них никогда не будет дополняющего пути. И их пометки можно никогда не очищать.

7. **Жадная инициализация не нужна**
   Если у вершины левой доли есть непокрытый сосед, то мы сможем добавить эту пару на первом проходе.
   Кроме того, эти вершины будут помечены как посещенные, так что первый проход не сможет ходить по чередующимся путям длины > 1 ребра. Значит, первый проход сделает ровно жадную инициализацию.

8. **Единственность паросочетания**
   Если их два, есть их симметрическую разность. Там есть чередующийся цикл или чередую-

щийся путь, который нельзя продлить ни в какую сторону.
Поищем чередующийся цикл или путь dfs-ом в стиле Куна.
Можно также заметить, что путь обязательно начинается в свободной вершине. И наоборот, если есть свободная, то и путь точно есть.

В недвудольном графе с поиском свободной вершины мы справимся, а вот цикл можем не найти.
Контрпример: ребра 1-2-3-4-1 и 1-3, в паросочетании 2-3 и 4-1. Если пойти `dfs` по пути 1-3-2, то мы уже никогда не найдем чередующийся цикл 1-2-3-4-1.
В двудольном это работало потому, что мы как бы ориентировали ребра из паросочетания справа налево, остальные слева направо, и надо было просто искать ориентирванный цикл.

Так что же делать в произвольном графе? Попробовать удалить каждое ребро, и поискать дополняющий путь без него. Если умеем за $T(V, E)$ искать дополняющий путь, то время $|M| \cdot T(V, E)$.
Умеют искать дополняющий путь за $\mathcal{O}(VE), \mathcal{O}(V^2), \mathcal{O}(E\alpha)$ (разными реализациями сжатия соцветий).
Сжатием соцветий можно искать сразу чередующийся цикл. В графе до сжатия соцветия есть чередующийся цикл $\Leftrightarrow$ он есть в графе после сжатия. Т.е. за $\mathcal{O}(E\alpha)$ при наличии max паросочетания можно проверить единственность.

9. **Разбиение на плавные подпоследовательности**
Сопоставим каждому элементу массива вершины l_i и r_i. Проведем все ребра (l_i, r_j), если j может быть в подпоследовательности после i, то есть $i < j$ и $|a_j - a_i| \leqslant X$.
Хотим для каждого элемента выбрать не более одного предыдущего и не более одного следующего, это как раз соответствует паросочетанию в таком графе. У скольких элементов нет следующего, столько и подпоследовательностей.
Min число подпоследовательностей $\Rightarrow$ min число непокрытых паросочетанием правых вершин (равное числу непокрытых левых) $\Rightarrow$ max паросочетание.

10. **Ездят такси, но нам нечем платить**
Сделаем заказы $(\texttt{from}, \texttt{to}, \texttt{time})$ вершинами графа. Ребро (a, b) проведем, если такси после запроса a может успеть обслужить запрос b, то есть $\texttt{time}_a + \texttt{dist}(\texttt{from}_a, \texttt{to}_a) + \texttt{dist}(\texttt{to}_a, \texttt{from}_b) \leqslant \texttt{time}_b$ (расстояния можно насчитать заранее). Граф ациклический, так как ребра ведут вперед по времени.
Теперь то же, что в предыдущей задаче, для каждого заказа хотим найти предыдущий и следующий, обслуживаемый той же машиной.

11. **Покрытие множества паросочетанием**
Пусть доли L, R. Запустим Куна из всех вершин множества A в любом порядке. Если покрылось, хорошо, иначе покрыть нельзя.
Почему? Запуск Куна из вершин только A равносилен запуску Куна на графе из долей (A, R), так как Кун никогда не смотрит на вершины левой доли, из которых его не запускали, ведь назад в левую долю мы ходим уже по ребрам из паросочетания.
Раз Кун корректен, он верно найдет паросочетание на графе из долей (A, R).

Если хотим max паросочетание, покрывающее все вершины из A, то же самое: просто сначала запустить обход Куна из всех вершин A, а потом уже из остальных.
Если обломаемся уже на A, то ответа нет, как мы поняли выше. Иначе всё получится, так

как алгоритм Куна никогда не лишает пары тех вершин, которым ее уже дал.

12. **Реберное покрытие**
    Ищем max паросочетание M, а каждую не покрытую им вершину покроем любым соседним с ней ребром.
    Размер полученного множества $|M| + (n - 2|M|) = n - |M|$.

    Корректность. Посмотрим на ребра оптимального ответа в каком-то порядке. Каждое ребро дает либо две новых вершины, либо одну.
    Ребра, дающие две новых вершины, образуют паросочетание M'. Тогда размер множества $|M'| + (n - 2|M'|) = n - |M'| \geqslant n - |M|$.

13. **Эйлеровость и паросочетания**
    Фактически, мы хотим найти d непересекающихся паросочетаний.
    Если d нечетное, найдем паросочетание, выкинем из графа.
    Если d четное, найдем эйлеров цикл (в каждой компоненте), разделим ребра по чередованию в цикле. Получим два графа степени $\frac{d}{2}$ на вершинах исходного. Рекурсивно покрасим оба графа.
    На каждом уровне рекурсии мы ищем паросочетание на графах из V вершин, а ребер во всех графах суммарно E. Если время поиска паросочетания таково, что $T(V, E_1) + T(V, E_2) \leqslant T(V, E_1 + E_2)$, то потратили $\mathcal{O}(\texttt{Matching} \cdot \log d)$. Для Куна ($\mathcal{O}(VE)$) и Хопкрофта-Карпа ($\mathcal{O}(E\sqrt{V})$) это верно.

14. **(*) Birkhoff–von Neumann decomposition**
    a) $\mathcal{O}(n^5)$. Построим полное паросочетание в графе, где доли – множество строк и множество столбцов, ребра – ненулевые клетки матрицы. $\mathcal{O}(n^3)$.
       Выберем минимальное по весу ребро x и вычтем из матрицы перестановочную подматрицу, заданную паросочетанием, с коэффициентом x.
       Из каждой строки и столбца вычлось ровно x, сумма осталась постоянной (не нужно, чтобы она была ровно 1). При этом занулилась хотя бы одно клетка, за $\leqslant n^2$ таких операций полностью разложим матрицу.

       Почему паросочетание найдется? По лемме Холла. У любых k строк сумма k, если у них $t < k$ столбцов-соседей, то у этих столбцов сумма хотя бы $k > t$.
    b) $\mathcal{O}(n^4)$. То же самое, но не ищем паросочетание с нуля, а достраиваем для строк с обнулившимися ребрами. То есть проход Куна за $\mathcal{O}(n^2)$ вызовется $\leqslant n^2$ раз.

15. **(*) Прямоугольные многоугольники**
    Нам нужно соединить с чем-то только вогнутые внутрь углы. Для каждого есть ровно два отрезка, которые можно провести. У некоторых углов какие-то отрезки могут совпадать.
    Двудольный граф: вертикальные и горизонтальные интересные отрезки – вершины, углы – ребра. Ищем минимальное вершинное покрытие.
    $\mathcal{O}(n)$ углов, $\mathcal{O}(n)$ отрезков, итого $\mathcal{O}(n^2)$.

Домашнее задание

3.1. Обязательная часть

1. **(1.5) Петя и гиперкуб**
   У маленького Пети любимая игрушка – d-мерный гиперкубик ($d \leqslant 10$). Некоторые вершины у куба белые, некоторые чёрные. Вы знаете цвета всех вершин, помогите Пете перекрасить как можно меньше белых вершин в красный цвет так, чтобы не было двух смежных по ребру белых вершин.

2. **(3) Лексмин вершинное покрытие**
   Дан двудольный граф. Среди всех вершинных покрытий минимального размера найти лексикографически минимальное. $\mathcal{O}(\text{poly}(V, E))$.

3. **(2+1) Разбиение на циклы**
   a) **(2)** Разбейте вершины **ор**графа на циклы. Каждая вершина должна быть покрыта ровно одним циклом. Либо скажите, что это невозможно.
   b) **(1)** Разбейте вершины **ор**графа с весами на ребрах на циклы min суммарного веса. Либо скажите, что это невозможно.

4. **(3) Степень не более двух**
   Дан произвольный **не**орграф. Найдите max по размеру мультимножество рёбер, такое, что степень каждой вершины $\leqslant 2$. Любое ребро можно брать два раза.

5. **(3) Множество прямых**
   Дано N различных прямых. Выбрать max по размеру подмножество прямых, такое, что никакие две прямые не параллельны, и никакие прямые не пересекаются в точке с $x = 0$.

6. **(3) Быстрое паросочетание в регулярном графе**
   Дан 2^k-регулярный двудольный граф. Найдите совершенное паросочетание за $\mathcal{O}(E)$ (детерминированным алгоритмом).

3.2. Дополнительная часть

1. **(3) Лексмин вершинное покрытие 2**
   Дан двудольный граф. Среди всех вершинных покрытий минимального размера найти лексикографически минимальное. $\mathcal{O}(VE)$.

2. **(3) Вершинно-взвешенное паросочетание**
   В двудольном графе у каждой вершины левой доли есть вес a_i. Вес ребра равен весу его конца из левой доли. Найдите паросочетание max веса за $\mathcal{O}(VE)$.
   Правильное решение без доказательства оценивается в 1 балл.

3. **(4) Вершинно-взвешенное паросочетание-2**
   В двудольном графе у каждой вершины левой доли есть вес a_i, у каждой вершины правой вес b_i. Вес ребра равен сумме весов его концов. Найдите паросочетание max веса за $\mathcal{O}(VE)$.
   Правильное решение без доказательства оценивается в 2 балла.

4. **(3) Чётность числа паросочетаний**
   Приведите полиномиальный алгоритм для нахождения чётности числа совершенных паросочетаний в двудольном графе с равным размером долей.

5. **Оптимизация H**
   Рассмотрим такую оптимизацию Куна.

```
bool kuhn(int v) {
  for (int u : graph[v])
    if (!used[u]) {
      used[u] = true;
      if (right_pair[u] == 0 || kuhn(right_pair[u])) {
        right_pair[u] = v;
        used[u] = false;
        return true;
      }
    }
  return false;
}
...
fill(used.begin(), used.end(), false);
for (int v = 1; v <= n; ++v) kuhn(v);
```

   Какой физический смысл происходящего?
   Оцените как-нибудь сверху и снизу асимптотику (в зависимости от $E, V, |M|$).
   Круто, если оценки сойдутся.
   Приведите тест (небольшой), на котором реализация работает некорректно.
   А можно ли ее завалить, если в начале сделать `random_shuffle` всех ребер?

   Задача исследовательская. Число баллов заранее не опеределено.