SPb HSE, 1 курс, зима 2024/25
Практика по алгоритмам #15

Первые сложности
21 января

Собрано 26 января 2025 г. в 16:26

Содержание

Первые сложности

1. **Циклы**
   a) Найти цикл в орграфе через данное ребро за $\mathcal{O}(E)$.
   b) Найти цикл в орграфе через данную вершину за $\mathcal{O}(E)$.
   c) Найти в неориентированом графе какой-нибудь цикл за $\mathcal{O}(E)$.
   d) Найти цикл в неорграфе через данное ребро за $\mathcal{O}(E)$.
   e) Найти цикл в неорграфе через данную вершину за $\mathcal{O}(E)$.
   f) Найти в неориентированом графе какой-нибудь цикл за $\mathcal{O}(V)$.

2. **Принцип домино**

   Потратьте, пожалуйста, время, чтобы без помощи практика осознать условие.

   На столе (на плоскости) стоят доминошки. Каждая доминошка стоит на ребре. Вася хочет по очереди уронить все доминошки на стол, каждую он может ронять в любую из двух сторон. Может ли Вася выбрать такой порядок и направления падения, чтобы ни в какой момент времени доминошки не касались друг друга?

3. **2-List-Coloring**

   2-List-Coloring: есть граф и k цветов, для каждой вершины дан список из двух цветов, в которые ее можно красить. Найти правильную покраску или сказать, что такой нет. $\mathcal{O}(V+E)$.

4. **3-List-Coloring**

   3-List-Coloring: есть граф и k цветов, но для каждой вершины дан список из трех цветов, в которые ее можно красить. Найти правильную покраску или сказать, что такой нет. $\mathcal{O}(1.5^V E)$.

5. **3-связность**

   Проверить граф на 3-связность
   a) вершинная за $\mathcal{O}(VE)$
   b) рёберная за $\mathcal{O}(VE)$

6. **Цикл длины 3**

   Дан неорграф. Найти в нём цикл длины ровно 3.
   a) $\mathcal{O}(VE/w)$
   b) (*) $\mathcal{O}(E\sqrt{E})$ (есть два решения!)
   c) (*) За $\mathcal{O}(V^{3-\varepsilon})$ при $E = \Theta(V^2)$

7. **Лежит ли задача в** NP**? А в** coNP **или** P**?**
   a) SUBSET-SUM (KNAPSACK без стоимостей).
   b) KNAPSACK со стоимостями.
   c) GI – проверка изоморфизма двух графов.
   d) Проверка *отсутствия* пути $a \rightsquigarrow b$ в графе.
   e) FACTOR$_d$ – наличие делителя не более d.
   f) Поиск кратчайшей эквивалентной записи булевой формулы
      1. Формула задана логическим выражением с `OR`, `AND`, `NOT`.
      2. Формула задана таблицей истинности.

8. **Перебор**

   Докажите, что NP $\subseteq$ EXP. Сперва по сути, за тем формально.

9. **Сведения**

   Постройте сведение для орграфа HAMPATH $\to$ HAMCYCLE.

10. **(*) Кактус, сэр!**

    Граф называется рёберным кактусом, если каждое ребро лежит не более чем на одном цикле. Дан граф, проверьте, является ли он кактусом, и, если да, найдите все его циклы. $\mathcal{O}(n+m)$.

11. **(*) Цикл длины 4**

    Дан неорграф. Найти в нём цикл длины ровно 4 за $\mathcal{O}(V^2)$.

12. **(*) Красно-синий турнир**

    Дан полный орграф. $\exists$ ребро $i \to j \Leftrightarrow i < j$. Каждое ребро красное или синее.
    Есть ли такая пара вершин a и b, что от a можно добраться до b как только по красным ребрам, так **И** только по синим ребрам? $\mathcal{O}(V+E)$.

13. **(*) Центры дерева**

    Дано дерево. Выбрать на нем k вершин так, чтобы сумма расстояний до ближайшей из выбранных вершин была минимальна.

Разбор задач практики

1. **Циклы**
   a) *Цикл в орграфе через ребро.* $e = (v, u)$, ищем путь из u в v.
   b) *Цикл в орграфе через вершину.*
   Запускаем dfs из v, цикл есть ⇔ мы нашли в процессе dfs ребро обратно в v.
   Способ #2. Удалим вершину, найдем путь из ее исходящих соседей в ее входящих соседей.
   Способ #3. Раздвоим вершину на v_{in} и v_{out}, все исходящие ребра проведем из v_{out}, все входящие в v_{in}, найдем путь из v_{out} в v_{in}.
   c) *В неорграфе ∀ цикл.* `dfs(v,parent)`, если нашли ребро в посещенную вершину, не равную `parent`, то цикл. Когда в `dfs(v,parent)` нужно пойти в вершину `u`, запускаем `dfs(u,v)`.
   d) *Цикл в неорграфе через ребро.* $e = (v,u)$, ищем путь из u в v, не проходя по e. Это просто запуск `dfs(u,parent=v)`.
   e) *Цикл в неорграфе через вершину.* Как в орграфе, но запрещаем dfs ходить в отца.
   f) *В неорграфе ∀ цикл за $\mathcal{O}(V)$.* В графе с V ребрами всегда есть цикл, считаем только первые V ребер.
   Но можно заметить, что стандартный способ с dfs тоже работает за $\mathcal{O}(V)$. Пока мы не нашли цикл, все посещенные dfs-ом ребра образуют дерево, итого мы упеем посмотреть как раз только на V ребер.

2. **Принцип домино**

   Задача сводится к `2-SAT`. x_i – в какую сторону падает i-я доминошка. Если прямоугольник $x_i = e_1$ пересекается с прямоугольником $x_j = e_2$, то делаем вывод $x_i \neq e_1 \wedge x_j \neq e_2$.

3. **2-List-Coloring**

   Задача сводится к `2-SAT`. У вершины v есть список c_v из двух цветов. Переменная $x_v = b$ означает «v покрашена в цвет $c_v[b]$».

   Если есть ребро $\{v, u\}$, то вершины v и u не могут быть покрашены в один цвет. Если $c_v[0] = c_u[0]$, записываем условие $x_v = 1 \vee x_u = 1$. И так в каждом из четырёх случаев:
   `forn(i, 2) forn(j, 2) if (c[v,i] == c[u,j]) ....`

   Если исходный граф состоял из V вершин и E рёбер, мы получили `2-SAT` из V переменных и $\leqslant 2E$ клозов.

   Напомним, что для `2-SAT` из n переменных и m клозов строится орграф из $2n$ вершин и $2m$ ребер, в котором за $\mathcal{O}(n + m)$ ищутся компоненты сильной связности.

4. **3-List-Coloring**

   Если запретить случайный цвет в каждой вершине, то получится `2-List-Coloring`, который решается за $\mathcal{O}(E)$.

   Если есть правильная покраска, то у каждой вершины мы запретили ее цвет с вероятностью $\leqslant \frac{1}{3}$. С вероятностью $\geqslant (\frac{2}{3})^V = p$ у всех вершин не запрещен нужный цвет.

   Получили решение за $\mathcal{O}(E)$, которое работает с вероятностью не менее p. Если повторить его $\frac{1}{p}$ раз, вероятность неудачи будет не более $(1-p)^{1/p} \approx e^{-1}$. Если повторить его $20\frac{1}{p} = \mathcal{O}(1.5^V)$ раз, вероятность неудачи будет не более $e^{-20} \approx 2 \cdot 10^{-9}$.

5. **3-связность**
   a) Вершинная. Нужно узнать, можно ли удалением двух вершин разрушить связность. Одну переберем, удалим, на оставшемся графе за $\mathcal{O}(E)$ проверяем наличие точек сочленения.
   b) Рёберная. Можно так же перебрать, выйдет $\mathcal{O}(E^2)$. Но ясно, что для разрушения связности нужно удалить хоть одно ребро из остовного дерева. Так что перебираем только $\mathcal{O}(V)$ ребер.

6. **Цикл длины 3**
   a) $\mathcal{O}(V^3)$: переберем все тройки вершин, проверим, что треугольник.
   b) $\mathcal{O}(VE)$: переберем ребро (z, x) и вершину y, проверим, что есть ребра $(x, y), (y, z)$.
   c) $\mathcal{O}(VE/w)$: переберем ребро (z, x). Вершина y ищется в пересечении `bitset`-ов соседей z и x.
   d) $\mathcal{O}(E\sqrt{E})$. Сначала, как в предыдущем пункте, переберем вершину x и ребро (y, z), но только при $\deg(x) > \sqrt{E}$. Таких x в графе не более $2\sqrt{E}$. Так нашли все треугольники, содержащие хоть одну вершину большой степени, за $\mathcal{O}(E\sqrt{E})$.
   Теперь найдем треугольники, в которых все вершины маленькой степени. Перебирем ребро (y, z), перебираем x – соседа z, смотрим, есть ли ребро (x, y). Если все степени x, y, x не более $\sqrt{E}$, мы нашли новый треугольник.
   Вместо этого можно перебрать ребро (z, x), перебрать y среди соседей вершины меньшей степени x, z, проверить наличие ребра во вторую. Сложность $\sum_{(z,x)} \min(\deg(x), \deg(z))$. Утверждается, что это $\mathcal{O}(E\sqrt{E})$.
   e) $o(V^3)$. Если возвести матрицу смежности графа в стпень k, то в ячейке $[u, v]$ будет число путей $u \rightsquigarrow v$ длины k. Пути не обязательно простые, могут повторяться вершин и ребра, но путь длины 3 из v в v – всегда треугольник.

   То есть надо посчитать A^3 за два умножения матриц, матрицы умеют умножать за $\mathcal{O}(n^{2.37})$. Потом за $\mathcal{O}(n^2)$ проверяем, что для какой-нибудь v $A^3[v, v] > 0$.

7. **Лежит ли задача в NP? А в coNP или P?**
   a) SUBSET-SUM. Подсказка для NP: набор предметов.
   b) KNAPSACK со стоимостями. NP – класс задач, для которых ответ true/false. Можно поменять условие задачи на «можно ли набрать цену больше C».
   Подсказка для NP: набор предметов. Решать за полином не умеют (динамика за $\mathcal{O}(nS)$ работает за экспоненту от длины S).
   c) GI. Подсказка для NP: перестановка, переводящая вершины первого графа в вершины второго. Умеют решать за $2^{\mathrm{poly}(\log n)}$. Полагают, что не P и не NPc.
   d) Отсутствие пути в P (dfs) $\Rightarrow$ в NP и в coNP (задачи из P решаются с пустой подсказкой).
   e) FACTOR$_d$. Подсказка для NP и coNP: разложение на простые.
   Чтобы проверить подсказку для coNP, нужно уметь проверять на простоту за полином от длины числа. Это умеют за n^6 ($n = \log N$ – длина числа).
   FACTOR умеют за $2^{Cn^{1/3}}$. Полагают, что не P и не NPc.
   f) Кратчайшая запись булевой формулы.
   Короткая запись – не подсказка: непонятно, как проверить ее эквивалентность исходной формуле. Не знают, лежит ли в NP.
   Если формула задана таблицей истинности (длина входа 2^n), то проверим эквивалентность этой таблицы T и короткой записи φ перебором всех подстановок переменных.

Время $2^n \cdot |\varphi| = \mathcal{O}(\text{poly}(2^n)) = \mathcal{O}(|T|) \Rightarrow$ в NP.
Если длина формулы φ больше $2^n n$, то она сразу не кратчайшая, ДНФ короче.

8. **Перебор**

   NP $\subseteq$ EXP: переберем подсказку.

   Почему длину подсказки y можно считать полиномиальной от длины входа x? Алгоритм проверки сертификата $\mathcal{A}(x, y)$ работает за время $\text{poly}(|x|)$, значит, успеет посмотреть лишь на $\text{poly}(|x|)$ бит подсказки y. Остальные биты y выкинем.

9. **Сведения**

   Пусть мы умеем искать цикл. Создадим фиктивную вершину, из неё во все, в неё из всех, найдём в новом графе цикл, удалим фиктивную вершину $\Rightarrow$ нашли в исходном графе путь.

10. **(*) Кактус, сэр!**

    Решение: dfs по неорграфу, который ищет циклы, а при нахождении цикла помечает все найденного цикла рёбра. Если какое-то ребро хочет пометиться дважды, это не рёберный кактус, брякаемся. Каждое ребро помечаем один раз $\Rightarrow \mathcal{O}(V + E)$.

    P.S. Чтобы проверять «вершинный кактус», действуем также, но помечаем вершины.

11. **(*) Цикл длины** 4

    Переберем пары ребер с общим концом (v, u), (v, w), пометим пару $\{u, w\}$, как имеющую общего соседа. Если нашли уже помеченную пару, то у этой пары два общих соседа, это цикл длины 4. Поскольку мы остановимся, как только хоть одна пара повторилась, время работы не больше, чем число различных пар, $\mathcal{O}(V^2)$.

12. **(*) Красно-синий турнир**

    Развернём все красные рёбра. Найдём цикл. Конец.

13. **(*) Центры дерева**

    Динамика по дереву $dp[v,i]$: ответ для поддерева v при условии, что сверху на расстоянии i будет вершина.

Домашнее задание

3.1. Обязательная часть

1. **(3) Провода**

   Есть n проводов. Есть круг из $2n$ разъемов, каждый разъем имеет тип от 1 до n (соответствующий номеру провода, который можно туда воткнуть), каждый тип встречается ровно два раза. У каждого провода есть цвет. В целях безопасности нельзя втыкать провода одинакового цвета в соседние разъемы. Найти способ соединить каждый из n проводов с одним из двух подходящих разъемов, не нарушив правила на цвета соседних.

2. **(2) Необходимые для достижимости ребра и вершины**

   Дан неорграф. Найти все рёбра и вершины, которые обязательно будут лежать на пути от a до b. $\mathcal{O}(V+E)$. (a) **(1)** найти все рёбра; (b) **(1)** найти все вершины.

3. **(2) Докажите утверждение**

   $\mathrm{P} = \mathrm{NP} \Rightarrow \mathrm{P} = \mathrm{coNP}$.

4. **(2) Сведения**

   Пусть мы знаем, что задача VERTEX-COVER (сказать, есть ли в графе множество вершин заданного размера, размер — часть входа, покрывающее все рёбра) лежит в классе NPc. Докажите, что задача CLIQUE (сказать есть ли в графе клика заданного размера, размер — часть входа) лежит в классе NPc.

3.2. Дополнительная часть

1. **(2) Полнота #4.** 0-1-ILP $\in$ NPc.

   ILP – целочисленное линейное программирование. Есть переменные $x_1, \ldots, x_n$.
   Даны неравенства $\langle a_i, x\rangle \leqslant b_i$ (скалярные произведения) и вектор c.
   Найти $x_i \in \{0, 1\}$, максимизирующие $\langle c, x\rangle$, удовлетворяющие всем неравенствам.
   Здесь a_i, x, b, c — вектора из нулей и единиц.

2. **(3) Необходимые для достижимости вершины в орграфе**

   Дан *орграф*.
   Найти все вершины, которые обязательно будут лежать на пути от a до b. $\mathcal{O}(V+E)$.

   Без строгого доказательства можно получить максимум половину от полного балла.

3. **(3) Удаление двух ребер**

   Нужно найти количество способов удалить из связного неорграфа два ребра так, чтобы он перестал быть связным.

   a) **(1)** $\mathcal{O}(V^2)$
   b) **(2)** $\mathcal{O}(E)$

3.3. Задачи, которые никто не будет проверять

1. **(3) Центры окружности**

   По кругу стоят n точек. Выбрать k из них так, чтобы минимальное расстояние между соседними выбранными было максимально. $\mathcal{O}(n \cdot \texttt{polylog})$.