# АМВ. ДЗ на неделю 9.

Прохоров Юрий, 771

## Задача 1

**Решение:**

Пусть $(u, v)$ — данное ребро, $a, b$ — данные вершины. Сделаем BFS из $a$. Найдем кратчайшие расстояния $d(u, a), d(a, b)$ и $d(v, a)$. Сделаем BFS из $b$ и найдем $d(u, b)$ и $d(v, b)$.

$$(u, v) \in \text{ какой-то кратчайший путь} \qquad \Longleftrightarrow \qquad \left[\begin{array}{l} d(u, a) + d(v, b) + 1 = d(a, b) \\ d(v, a) + d(u, b) + 1 = d(a, b) \end{array}\right.$$

Асимптотика совпадает с асимптотикой BFS и равна $O(|V| + |E|)$. Докажем корректность.

• Пусть $(u, v)$ лежит на кратчайшем пути. Без ограничения общности, пусть из $a$ путь сначала приходит в $u$:

$$a \to \ldots \to u \to v \to \ldots \to b$$

Так как любой подпуть кратчайшего пути, тоже кратчайший, то выполняется равенство:

$$d(a, u) + 1 + d(v, b) = d(a, b)$$

• Пусть выполняется равенство выше. Тогда путь из $a$ в $b$, содержащий $(u, v)$ будет состоять из путей $a \to u$ и $v \to b$, на которых достигаются оценки $d(a, u)$ и $d(v, b)$ соответственно, и самого ребра $(u, v)$.

## Задача 2

**Решение:**

Пусть надо найти путь из $a$ в $b$.

Выполним процедуру СТЯГИВАНИЕ для всех нулевых ребер в графе $G$. Запустим BFS в графе с мультивершинами (в нем все ребра единичные) из $a$ и найдем кратчайший путь в $b$. Затем будем идти по этому пути и разворачивать мультивершины. Пусть мы пришли в мультивершину $M$ по ребру $e_1$, а вышли по ребру $e_2$. Эти ребра были в исходном графе, поэтому мы знаем из какой вершины $(v_1)$ мы пришли в $M$, и в какую вышли $(v_2)$. Запустим в графе $M$ с добавленными ребрами $e_1$ и $e_2$ и вершинами $v_1, v_2$ BFS из $v_1$. Найдем кратчайший путь из $v_1$ в $v_2$. Это и будет нужный кусок пути через данную мультивершину.

На стягивание требуется $O(|E| + |V|)$ времени (можно делать с помощью BFS, находя компоненты связности с точки зрения нулевых ребер). Затем нужно делать много $BFS$ для мультивершин. Но суммарное число ребер и вершин в них меньше, чем ребер и вершин в исходном графе, поэтому в сумме все будет работать за $O(|E| + |V|)$.

Покажем корректность. Веса кратчайшего пути из $a$ в $b$ в исходном и стянутом графах совпадают, так как мы убрали только нулевые ребра. В силу корректности $DFS$, мы найдем кратчайший путь из $a$ в $b$ в стянутом графе. Затем мы добавим в него нулевые ребра, чтобы получить путь в исходном графе. Его длина такая же, значит, это и есть нужный кратчайший путь.

## Задача 3

**Решение:**

# Задача 4

**Решение:**

**(a)** В алгоритме Флойда-Уоршелла $d^{(k)}$ — матрица кратчайших расстояний таких, что пути могут проходить только через промежуточные вершины с номерами $\leq k$. Поэтому $d^{(n)}$ — все кратчайшие пути, если нет отрицательных циклов.

Если нет отрицательных циклов, то на главной диагонали $d^{(n)}$ стоят нули, иначе — где-то будет отрицательное число. Поэтому ток можно обнаруживать отрицательные циклы.

**(б)** Алгоритм Флойда-Уоршелла:

$$d^{(0)} = \begin{pmatrix} 0 & 1 & 3 & -1 \\ \infty & 0 & \infty & -2 \\ \infty & -2 & 0 & -4 \\ \infty & \infty & \infty & 0 \end{pmatrix}$$

$$d^{(1)} = \begin{pmatrix} 0 & 1 & 3 & -1 \\ \infty & 0 & \infty & -2 \\ \infty & -2 & 0 & -4 \\ \infty & \infty & \infty & 0 \end{pmatrix} \qquad - \text{ такая же, так как в вершину 1 не идут ребра}$$

$$d^{(2)} = \begin{pmatrix} 0 & 1 & 3 & -1 \\ \infty & 0 & \infty & -2 \\ \infty & -2 & 0 & -4 \\ \infty & \infty & \infty & 0 \end{pmatrix} \qquad - \text{ то же самое}$$

$$d^{(3)} = \begin{pmatrix} 0 & 1 & 3 & -1 \\ \infty & 0 & \infty & -2 \\ \infty & -2 & 0 & -4 \\ \infty & \infty & \infty & 0 \end{pmatrix} \qquad - \text{ то же самое}$$

$$d^{(3)} = \begin{pmatrix} 0 & 1 & 3 & -1 \\ \infty & 0 & \infty & -2 \\ \infty & -2 & 0 & -4 \\ \infty & \infty & \infty & 0 \end{pmatrix} \qquad - \text{ то же самое, так как из вершины 4 нет ребер}$$

# Задача 5

**Решение:**

**(b)** Итерация, на которой перестанут обновляться веса в алгоритме Форда-Беллмана, сильно зависит от порядка обхода ребер графа. В любом случае, если длина максимального кратчайшего пути из стартовой вершины равна $k$, то после $k$-ой итерации расстояния перестанут меняться. Докажем это.

| # | $A$ | $B$ | $C$ | $D$ | $E$ | $F$ | $G$ | $H$ |
|---|---|---|---|---|---|---|---|---|
| 0 | $\infty$ | $\infty$ | $\infty$ | 0 | $\infty$ | $\infty$ | $\infty$ | $\infty$ |
| 1 | $\infty$ | $\infty$ | 7 | | $\infty$ | $\infty$ | $\infty$ | 14 |
| 2 | $\infty$ | 15 | | | 9 | $\infty$ | $\infty$ | 11 |
| 3 | $\infty$ | 15 | | | | 16 | 15 | 11 |
| 4 | $\infty$ | 15 | | | | 16 | 13 | |
| 5 | $\infty$ | 15 | | | | 14 | | |
| 6 | 22 | 15 | | | | | | |
| 7 | 19 | | | | | | | |
| parent | $B$ | $C$ | $D$ | | $C$ | $G$ | $H$ | $C$ |

На каждой итерации мы перебираем релаксируем по всем ребрам, поэтому на $i$-ой итерации мы рассмотрим $i$-ое ребро каждого кратчайшего пути. Значит, после $k$ итераций все кратчайшие пути будут найдены, и мы ничего обновляться не будет.

В приведенном графе длину максимального кратчайшего пути можно найти с помощью, например, алгортима Дейкстры (таблица выше). Максимальный кратчайший путь состоит из 4 ребер:

$$D \to C \to H \to G \to F$$

поэтому расстояния в алгоритме Флойда-Уоршелла перестанут меняться после 4 шага.

**(c)** Будем понимать условие следующим образом: можно подобрать такой порядок перебора ребер, что расстояния обновляются на каждой из $n-1$ фаз, где $|V| = n$.

Искомым множеством является множество графов, в которых есть кратчайший путь длины $n-1$. Так как все ребра единичные, то под длиной пути можно понимать реберную длину. Путь проходит через все $n$ вершин, и является кратчайшим. Занумеруем вершины от 1 до $n$. Так как любой подпуть кратчайшего пути — тоже кратчайший, то в графе нет других ребер, кроме тех, что в этом пути. Значит, с точностью до изоморфизма такой граф единственный:

$$1 \to 2 \to 3 \to \ldots \to n-1 \to n$$

Будем обходить ребра справа налево. Тогда на каждой последующей фазе что-то будет обновляться.

Доказательство в обратную сторону аналогично предыдущему пункту: пусть все $n-1$ фаз что-то обновляется, что максимальный кратчайший путь содержит $< n-1$ ребер. Но такого все перестанет обновляться после меньшего числа фаз — противоречие.

## Задача 6

**Решение:**

Пусть $\leq$ — нестрогое отношение частичного порядка, $<$ — соответствующее ему строгое, согласно этому определению:

> **Определение 5.** Бинарное отношение $\leq$ называется отношением нестрогого частичного порядка (или просто частичным порядком), если $a \leq b \Leftrightarrow (a < b)$ или $(a = b)$.

Допустим в графе есть цикл. Тогда

$$a_1 \leq a_2 \leq a_3 \leq \ldots \leq a_n \leq a_1$$

В силу того, что все соседние элементы различны:

$$a_1 < a_2 < a_3 < \ldots < a_n < a_1 \qquad \Longrightarrow \qquad a_1 < a_1$$

А это противоречит определению строгого отношения частичного порядка. Значит, в графе нет циклов. Пусть есть компонента сильной связности из более, чем 1-ой вершины. Тогда в графе есть цикл хотя бы на двух вершинах. Итак, каждая вершина графа — своя компонента сильной связности. Тогда в конденсации графа будет столько же вершин, как и в исходном графе, то есть $|V|$ вершин.

## Задача 7

**Решение:**

**(a)** Обозначим $\text{in}(x)$ и $\text{out}(x)$ — времена входа и выхода для вершины $x$ при DFS. Достижимость $y$ из $x$ обозначим так: $x \mapsto y$.

Требуется доказать, что

$$u \mapsto v,\ v \not\mapsto u \qquad \Longrightarrow \qquad \text{out}(u) > \text{out}(v)$$

Рассмотрим два случая.

• $\text{in}(u) < \text{in}(v)$. Тогда DFS зайдет в $v$, перед тем как выйти из $u$, так как $v$ достижима из $u$, а в $v$ мы еще не заходили. Затем DFS исследует все достижимые из $v$ вершины (в них не входит $u$) и выйдет из $v$. Только потом он выйдет из $u$.

• $\text{in}(u) > \text{in}(v)$. Тогда мы исследуем все достижимые из $v$ вершины перед выходом из $v$. Среди них нет $u$, поэтому в $u$ DFS может войти только после выхода из $v$.

**(b)** Допустим противное. Пусть существует ребро $(u, v)$ такое, что $\text{out}(v) > \text{out}(u)$. Граф ациклический, значит, $v \not\mapsto u$, иначе будет цикл. По утверждению из предыдущего пункта,

$$\text{out}(u) > \text{out}(v)$$

Это противоречие.

## Задача 8

**Решение:**

**(a)** Компоненты сильной связности — классы эквивалентности относительно отношения взаимной достижимости. Поэтому если мы покажем, что в инвертированном графе эти отношения сохраняются, то и в классы эквивалентности сохраняются. Пусть $G$ — исходный граф, $G'$ — инвертированный граф.

$$G: \ u \mapsto v, \ v \mapsto u$$

Тогда существуют соответствующие пути: $l_1$ : из $u$ в $v$, и $l_2$ : из $v$ в $u$. В $G'$ путь $l_1$ станет путем из $v$ в $u$, а $l_2$ — обратно. Поэтому отношение взаимной достижимости сохраняется.

Пусть теперь

$$G: \ u \not\mapsto v \ \vee \ v \not\mapsto u$$

Предположим, что

$$G': \ u \mapsto v, \ v \mapsto u$$

Но тогда по уже доказанному в $G'' \equiv G$ будет взаимная достижимость — противоречие.

**(b)** Алгоритм состоит из двух поисков в глубину. На сортировку вершин не тратится время, так как DFS может возвращать вершины в нужном порядке. Итого асимптотика:

$$O(|V| + |E|).$$

Нужно только $O(|V|)$ дополнительной памяти, в которой мы храним упорядоченные во времени выхода вершины после первого DFS.

**(c)** Верно, потому что конденсация — это ориентированный ациклический граф (если бы был цикл, то можно было бы стянуть некоторые компоненты сильной связности). Из задачи 7 следует, что для таких графов можно проводить топологическую сортировку.