# Решения дополнительных задач по курсу «АиСД»

Лунга Артём

**Задача 1.** Найти количество треугольников в графе за $O(E\sqrt{E})$

**Решение.** Назовём вершину *общительной*, если у неё не меньше, чем $\lceil E\rceil$ соседей. Иначе будем говорить, что вершина *необщительная*.

В графе не более $2 \cdot \lceil\sqrt{E}\rceil$ общительных вершин, так как иначе мы бы получили, что в графе больше, чем $\frac{2\cdot\lceil\sqrt{E}\rceil\cdot\lceil\sqrt{E}\rceil}{2}$, то есть больше, чем $E$ рёбер.

Количество треугольников, которые построены на общительных вершинах не превосходит $C^3_{2\cdot\lceil\sqrt{E}\rceil}$. То есть равно $O(E\sqrt{E})$.

Пусть теперь треугольник содержит хотя бы одну необщительную вершину. В таком треугольнике будет хотя бы два ребра, которые инцидентны необщительной вершине. Всего таких пар рёбер $O(E)$, причём *парными* для каждого ребра могут быть $O(\sqrt{E})$ рёбер, так как степень необщительной вершины меньше $\lceil\sqrt{E}\rceil$. Значит, треугольников, которые содержат необщительные вершины $O(E\sqrt{E})$.

Таким образом, всего треугольников в графе $O(E\sqrt{E})$.

Так как асимптотики шагов мы доказали, опишем алгоритм.

Общительная вершина или нет мы можем узнать на этапе ввода данных, например, с помощью хеш-таблицы(номер вершины — количество соседей). Таким же образом мы можем задать некоторые отметки на рёбрах, которые инцидентны необщительным вершинам.

Для общительных вершин проверяем образуют ли они треугольники. Это занимает $O(E\sqrt{E})$.

Для отмеченных рёбер будем искать парные(в смысле оговорённом ранее) и проверять образуют ли они треугольник. Это занимает $O(E\sqrt{E})$.

Таким образом, работа всего алгоритма займёт $O(E\sqrt{E})$.

**Задача 2.** Ориентировать неориентированный граф так, чтобы он стал сильно-связным за $O(V + E)$ или сказать, что это невозможно.

**Решение.**

Проверим есть ли графе изолированные вершины. Сделаем это за $O(V)$, посмотрев степени всех вершин. Если есть хотя бы одна, то нельзя ориентировать граф так, чтобы он стал сильно-связным.

Найдём с помощью обхода в глубину за $O(V+E)$ количество компонент связности в графе. Если их хотя бы две, то нельзя ориентировать граф указанным образом.

Проверим с помощью поиска в глубину за $O(V + E)$ есть ли в графе мост. Если мы его обнаружили, то не получится ориентировать граф так, чтобы он стал сильно-связным, так как мы сможем по мосту пройти из одного блока в другой, но вернуться уже не сможем.

Если мостов в графе нет, то выполним обход в глубину из некоторой вершины $v$. Все рёбра, которые мы проходим во время обхода в глубину и которые ведут в ещё не пройденную вершину, ориентируем от $v$. Оставшиеся рёбра — по направлению к $v$.

Из вершины $v$ можно попасть в любую другую вершину графа, двигаясь по рёбрам в соответствующем направлении. Так как в графе нет мостов, то обязательно существует хотя бы две вершины смежные с $v$, причём по построению в одну из них ведёт ребро из $v$, а из другой исходит ребро в $v$.

Допустим, мы хотим попасть из некоторой вершины $u$ в вершину $w$. Если при запуске обхода в глубину мы заходили в $u$ раньше, чем в $w$, то построим путь из $v$ в $w$ и исключим дуги, которые содержатся в пути из $v$ в $u$. Иначе построим путь из $w$ в $v$(мы всегда можем сделать это, так как есть вершина, ребро из которой ведёт в $v$), после построим путь из $v$ в $u$.

Таким образом, если неориентированный граф удовлетворяет всем требуемым свойствам, то мы можем за $O(V+E)$ ориентировать его так, чтобы он стал сильно-связным.

**Задача 3.** Разбить все ребра неориентированного графа на минимальное число путей за $O(V+E)$.

**Решение.**

С помощью обхода в глубину за $O(V+E)$ найдём все компоненты связности графа. Будем обрабатывать каждую из них отдельно, поэтому далее без ограничения общности проведём рассуждения для связного графа.

Так как граф является связным, то мы можем дополнить его рёбрами до Эйлерового. В графе количество вершин нечётной степени чётно, поэтому мы можем выделить непересекающиеся пары вершин нечётной степени и провести внутри этих пар ребро.

С помощью обхода в глубину за $O(V+E)$ найдём Эйлеров цикл. После удалим рёбра, которые добавили, тогда цикл распадётся на пути. Путей будет в точности столько, сколькими рёбрами мы дополняли граф до Эйлерова.

Кроме того, если у нас заданы пути, то мы можем их по циклу соединить и получить Эйлеров цикл. Значит, задача минимизации числа путей эквивалентна задаче о минимизации числа рёбер, которые надо добавить, чтобы граф стал Эйлеровым. Поэтому описанный нами алгоритм корректен: он находит минимальное число путей и делает это за $O(V+E)$.