Свободное перемещение

Автор задачи: Даниил Орешников, разработчик: Константин Бац

Задачу можно переформулировать как «ориентировать ребра неориентированного графа, чтобы максимизировать количество путей длины 2». Ключевой идеей для решения задачи являлось наблюдение, что количество путей длины 2 в графе в точности равно следующей сумме:

$$\sum_{v=1}^{n} \mathtt{deg}_{in}(v) \cdot \mathtt{deg}_{out}(v),$$

где $\mathtt{deg}_{in}(v)$ и $\mathtt{deg}_{out}$ — входящая и исходящая степени вершины v соответственно. Действительно, переберем «центральную» вершину пути, и заметим, что независимо друг от друга первую вершину можно выбрать $\mathtt{deg}_{in}$ способами, а третью — $\mathtt{deg}_{out}$ способами.

Для решения первой подзадачи можно было сделать полный перебор того, в какую сторону ориентировать каждое ребро, после чего посчитать для каждого способа указанную выше сумму. Такое решение работает за время $\mathcal{O}(2^n \cdot n)$.

Вторая подзадача решается даже без использования ключевой идеи. Граф, в котором степень каждой вершины не превосходит двух, разбивается на изолированные вершины, пути и циклы. Интуитивно очевидно, что количество путей длины 2 максимально тогда, когда ребра каждого пути и цикла ориентированы в одном направлении (то есть когда после ориентации ребер пути все еще являются путями, а циклы — циклами). Ориентировать ребра таким образом можно за один проход поиска в глубину за время $\mathcal{O}(n)$.

Для решения следующих подгрупп достаточно заметить, что $\mathtt{deg}_{in}(v) + \mathtt{deg}_{out}(v)$ равно $\mathtt{deg}(v)$, то есть степени вершины в исходном графе. При фиксированной сумма произведение двух величин тем больше, чем ближе эти две величины друг к другу. Таким образом, максимум достигается тогда, когда $|\mathtt{deg}_{in}(v) - \mathtt{deg})_{out}(v)|$ минимально, то есть равно 0 или 1 (в зависимости от четности степени вершины).

В третьей подзадаче есть конструктивный способ ориентировать ребра графа так, чтобы входящие степени были максимально близки к исходящим. Мысленно расположим все вершины графа по кругу и ориентируем ребра из вершины v в $\left\lfloor \frac{n}{2} \right\rfloor$ ближайших по направлению по часовой стрелке, тогда автоматически ребра из ближайших против часовой стрелки будут направлены из них в v. Останутся только ребра между противоположными по кругу вершинами, если n нечетно, их можно ориентировать в любую сторону. Время работы решения равно $\mathcal{O}(m)$.

Ориентировать дерево в четвертой подзадаче подобным образом тоже можно конструктивно: подвесим дерево за корень и будем спускаться от него рекурсивно вниз. Ориентируем ребра в половину детей вниз, а из половины детей — вверх. Если у корня нечетное число детей, ребро в оставшегося ребенка можно ориентировать в любую сторону.

Затем рекурсивно запустимся из всех детей. Если у текущей вершины четное число своих детей, ориентируем ровно по половине из них в каждую сторону. Если же нечетное — смотрим на ребро в саму текущую вершину сверху, если оно ориентировано в нее, то «лишнее» ребро надо ориентировать вниз в ребенка, и наоборот. Время работы решения равно $\mathcal{O}(n)$.

Для полного решения можно было разбить все вершины с нечетной степенью на пары, и провести в этих парах ребра. После такого действия все степени вершин станут четными, а значит в графе будет Эйлеров цикл — цикл, проходящий по каждому ребру ровно один раз. Запустим алгоритм поиска Эйлерова цикла (это просто поиск в глубину), и ориентируем ребра в том направлении, в котором по ним пройдем.

Теперь, благодаря свойствам Эйлерова цикла, выполняется $\mathtt{deg}_{in}(v) = \mathtt{deg}_{out}(v)$ для каждой вершины v. Если удалить те ребра, которые мы изначально добавили, степень каждой вершины изменится не более, чем на 1, таким образом входящие и исходящие степени каждой вершины максимально близки. Время работы решения, как и до этого — $\mathcal{O}(m)$.