

Безопасное мореплавание

Переформулируем задачу. Дан неориентированный граф, требуется для каждого ребра определить максимальный его вес, при котором оно будет входить в минимальное остовное дерево.

Задача разбивается на несколько похожих случаев. Проще всего начать со случая ребер, которые не входят в MST. Для таких ребер достаточно заранее построить произвольное MST, после чего воспользоваться леммой о безопасном ребре: чтобы ребро входило в MST, оно должно быть одним из минимальных в каком-то разрезе графа, разделяющем концы этого ребра.

Иными словами, построим MST и обозначим множество ребер в нем M . Рассмотрим какое-то ребро $e \notin M$. Если добавить это ребро в MST, мы получим множество $M \cup \{e\}$, в котором есть цикл, проходящий по e . Чтобы существовал минимальный остов, содержащий e , как следствие из леммы о безопасном ребре, необходимо и достаточно, чтобы на цикле существовало ребро веса не меньше p_e . Таким образом, ответ для $e \notin M$ — это вес максимального ребра на образующемся цикле, который можно найти с помощью двоичных подъемов и `lca` за время $\mathcal{O}(\log n)$.

В случае с ребрами $e \in M$ все аналогично: до какого-то момента можно просто увеличивать вес ребра, но в какой-то момент появится «замена», которую можно добавить вместо e , уменьшив вес дерева. Пусть при удалении e дерево M разбивается на M_1 и M_2 . Тогда заметим, что $p_e = \min_{u \in M_1, v \in M_2, (uv) \notin M} w_{uv}$ — это вес минимального ребра между M_1 и M_2 , на которое можно заменить удаляемое e . При большем весе e эта замена приводит к уменьшению веса остова, что означает, что e никакому минимальному остову не принадлежит.

Осталось эту величину посчитать для каждого $e \in M$. Для этого есть два способа:

- Проще всего ее посчитать подобием динамики по дереву: будем для каждой v поддерживать множество ребер, ведущих из поддерева v в «над-дерево» parent_v . Для этого необходимо идти по дереву снизу вверх, и для каждой v объединять такие множества для детей, после чего удалять все ребра, смежные с v .

При таком алгоритме каждое ребро будет ровно один раз добавлено в какое-то множество и ровно один раз удалено, поэтому в сумме это займет $\mathcal{O}(m \log m)$ времени. А объединения множеств с помощью техники `small-to-large` займут в сумме $\mathcal{O}(m \log^2 m)$ времени.

- Возможно, решение за такую асимптотику не будет проходить по времени — тогда достаточно заметить, что любое ребро $g \notin M$ может являться «заменой» для каждого ребра на пути в M между его концами. То есть вместо того, чтобы для каждого $e \in M$ считать $p_e = \min_{u \in M_1, v \in M_2, (uv) \notin M} w_{uv}$, будем рассматривать $(uv) \notin M$ и выполнять обновление p_e для всех e на пути в M между v и u . Такие обновления можно выполнять с помощью отложенной информации на тех же самых двоичных подъемах, за суммарное время $\mathcal{O}(m \log m)$.