Разбор задачи «Операция «Перестановка»»

Автор задачи:	Антон Гардер
Подготовка условия, решения и тестов:	Василий Алферов
Автор разбора:	Василий Алферов

Сделаем двоичный поиск по ответу — количеству утверждений, после которых можно однозначно восстанавливать исходную перестановку. Двоичный поиск можно применить, так как если перестановку можно восстановить после x утверждений, то тем более её можно однозначно восстановить после $x + 1$ утверждения.

Посмотрим, как проверить, что после k утверждений можно восстановить требуемую перестановку. Построим граф с n вершинами, где i-я вершина — это позиция солдата в перестановке. Пусть известно, что номер солдата, стоявшего в a-й позиции больше номера солдата, стоявшего в b-й позиции. Тогда проведём ориентированное ребро из вершины b в вершину a. Сделаем так для первых k утверждений.

Заметим, что граф получится ациклическим даже для $k = m$, так как по условию гарантируется, что существует хотя бы одна перестановка, для которой выполняются все условия, которые помнит адъютант. У ациклического графа есть топологическая сортировка — такой порядок вершин, что ребра идут только из вершин с меньшими номерами в вершины с большими. Топологическую сортировку графа с V вершинами и E ребрами можно построить с помощью обхода в глубину за время $O(V + E)$.

Заметим, что есть взаимно-однозначное соответствие между топологическими сотрировками получившегося графа и подходящими построениями содлат. Действительно, пусть задана топологическая сортировка, тогда для обратной перестановки выполняются все условия: если на a-й позиции должен стоять солдат с большим номером, чем на b-й, то из b в a проведено ребро, значит, b в обратной перестановке находится раньше, а значит, номер солдата, стоящего на b-м месте меньше. Обратно, если у нас есть какой-нибудь ответ, обратная ему перестановка соответствует какой-то топологической сортировке графа. Итак, для того, чтобы проверить единственность возможного построения, необходимо и достаточно проверить единственность топологической сортировки в этом графе.

Это можно сделать следующим образом. Построим какую-нибудь топологическую сортировку и проверим, что между каждой парой соседних вершин в ней есть ребро. Если это условие выполняется, то никакой другой топологической сортировки быть не может. Обратно, если топологическая сортировка единственна, то между каждой парой соседних вершин есть ребро, иначе можно поменять местами пару соседних вершин, между которыми ребра нет, и получить другую топологическую сортировку.

Сложность такого решения $O((n + m) \log m)$, оно укладывается в ограничения по времени.

Однако можно усовершенствовать это решение и получить решение за $O(n + m)$. Заметим, что если ответ не равен -1, то в графе, построенном на всех запросах, топологическая сортировка единственна. Заметим также, что если она в какой-то момент стала единственной, то при добавлении ребер уже не поменяется. Значит, достаточно топологически отсортировать итоговый граф и взять в качестве ответа максимальный номер ребра между соседними вершинами.