Kaleidoscopic Route

Идея: Михаил Дворкин
Разработка: Павел Маврин

Для начала давайте построим граф кратчайших путей из вершины 1 в вершину n. Для этого два раза запустим обход в ширину: из вершины 1 и из вершины n, и оставим в графе ребра (u, v), для которых $d(1, u) + d(n, v) + 1 = d(1, n)$, ориентировав их от u к v. Получившийся ациклический ориентированный граф содержит все кратчайшие пути из 1 в n, и только их (то есть любой путь из 1 в n в этом графе — кратчайший). Теперь в этом графе нам нужно найти путь с максимальной разницей между максимальным и минимальным ребром.

Пусть кратчайший путь состоит из более чем одного ребра (если путь состоит из одного ребра, максимальная разница очевидно 0), тогда максимальное и минимальное ребро — это какие-то два различных ребра на этом пути. Пусть, например, мы сначала встретили минимальное ребро, а затем максимальное. Рассмотрим любую вершину на нашем пути между этими ребрами, пусть это вершина v, тогда путь делится на два отрезка: от 1 до v, и от v до n. Чтобы ответ был оптимальным, нужно чтобы на пути то 1 до v минимальное ребро было как можно меньше, а на пути от v до n максимальное ребро было как можно больше.

С помощью динамического программирования можно для каждой вершины v найти путь из 1 в v с наименьшим возможным минимальным ребром: $d_1[v] = \min(\min(d_1[u], c_{uv}))$. Аналогично можно найти для каждой вершины путь из v в n с наибольшим максимальным ребром: $d_2[v] = \max(\max(d_2[u], c_{vu}))$. Теперь найдем вершину с максимальной разностью $d_2[v] - d_1[v]$, и восстановим от нее соответствующие отрезки пути.

Аналогично найдем оптимальный путь, в котором сначала встречается максимальное ребро, а потом минимальное. Из этих двух ответов выберем наилучший.