Исследование операций

Лабораторная работа 5 «Задача о наидлиннейшем пути из заданной вершины в заданную вершину в направленном графе без контуров»

Неориентированным графом называется упорядоченная пара множеств $G = (V, E)$, в которой множество V — это конечное непустое множество, элементы которого называются вершинами, и множество E — это множество неупорядоченных пар различных вершин. При этом неупорядоченные пары из множества E называются ребрами графа G.

Ребро неориентированного графа — это неупорядоченная пара различных вершин. Рассмотрим ребро $e = \{u, v\} \in E$ с концевыми вершинами u, v. Ребро можно ориентировать двумя способами. Что значит ориентировать ребро e? Ориентировать ребро e — это значит указать какая из двух концевых вершин ребра e первая, а какая вторая. При этом ориентированное ребро (ребро с ориентацией) называется дугой, первая вершина дуги называется ее началом, а вторая вершина — ее концом.

Если задать ориентацию каждому ребру неориентированного графа, то получится направленный граф.

Определение 1. Пусть $G = (V, A)$ — направленный граф. Чередующаяся последовательность вершин $v_i \in V$ и дуг $a_j \in A$

$$v_0\ a_1\ v_1\ a_2\ v_2\ \ldots\ v_{k-1}\ a_k\ v_k\ \ldots\ v_{n-1}\ a_n\ v_n,$$

в которой каждой дуге a_k предшествует ее начало v_{k-1}, а следует за дугой ее конец v_k

$$\forall\, k \in \{1, 2, \ldots, n\} \quad a_k = (v_{k-1}, v_k) \in A$$

называется (v_0, v_n)-маршрутом.

Если в графе существует (v_0, v_n)-маршрут, то мы говорим, что вершина v_n достижима из вершины v_0. Классифицируем маршруты следующим образом:
(v_0, v_n)-маршрут называется открытым, если $v_0 \neq v_n$,
(v_0, v_n)-маршрут называется замкнутым, если $v_0 = v_n$.

Определение 2. Открытый (v_0, v_n)-маршрут, в котором нет повторяющихся дуг, называется (v_0, v_n)-цепью. В свою очередь, (v_0, v_n)-цепь, в которой нет повторяющихся вершин, называется (v_0, v_n)-путем. Замкнутый (v_0, v_n)-маршрут, в котором нет повторяющихся дуг, называется циклом. Цикл, в котором нет повторяющихся вершин, кроме крайних, называется контуром.

Задан направленный граф $G = (V, A)$ без контуров. Каждой дуге графа приписано целое неотрицательное число — длина дуги

$$\ell : A \to \{z \in \mathbb{Z} : z \geqslant 0\}.$$

В графе G выделены две вершины s и t. Требуется определить достижима ли в графе G вершина t из вершины s и в случае положительного ответа найти наидлиннейший (s, t)-путь. Длина (s, t)-пути — это сумма длин его дуг.

Рассмотрим алгоритм динамического программирования, решающий эту задачу. Хорошо известно, что вершины любого направленного графа без контуров можно топологически упорядочить, т.е. расположить вершины графа последовательно друг за другом таким образом, чтобы все дуги графа располагались слева направо.

Пусть вершин графа занумерованы в порядке их следования в топологической сортировке $V = \{v_1, v_2, \ldots, v_n\}$. Пусть в топологическом упорядочении вершин графа G выделенная вершина s идет k-ой по счету, а выделенная вершина t — ℓ-ой по счету. Таким образом, $s = v_k$ и $t = v_\ell$. Предполагается, что вершины s и t различные. Стало быть, $k \neq \ell$.

Пусть $k > \ell$, т.е. в топологическом упорядочении вершин вершина s правее вершины t: $\ldots\ t\ \ldots\ s\ \ldots$. Все дуги «идут» слева направо. Поэтому все пути, выходящие из вершины s «идут вправо» и не могут достичь вершину t. Ясно, что в этом случае вершина t недостижима из вершины s.

Пусть $k < \ell$. Для того, чтобы найти наидлиннейший путь из k-ой вершины в ℓ-ую вершину достаточно рассмотреть только вершины от k-ой до ℓ-ой включительно

$$\begin{array}{ccccccccc} \bigcirc & \bigcirc & \bigcirc & \cdots & \bigcirc & \cdots & \bigcirc & \bigcirc \\ s = v_k & v_{k+1} & v_{k+2} & & v_i & & v_{\ell-1} & v_\ell = t \end{array}$$

Мы исключаем из рассмотрения все вершины до k-ой, а также все вершины после ℓ-ой вершины. Будем проходить все вершины в том порядке, в котором они идут в топологическом упорядочении, начиная с вершины v_k. Для каждой вершины v_i, $i \in \{1, 2, \ldots, k\}$, найдем максимальную длину (v_k, v_i)-пути, которую обозначим через $OPT(v_i)$. Найдем максимальную длину пути от вершины v_k до вершины v_k, затем до вершины v_{k+1}, после чего до вершины v_{k+2} и т.д.

Рассмотрим вершину v_k. Ясно, что

$$OPT(v_k) = 0. \tag{1}$$

Пусть для всех вершин $v_k, v_{k+1}, \ldots, v_{i-1}$ мы нашли максимальную длину пути, т.е. значения $OPT(v_k), OPT(v_{k+1}), \ldots, OPT(v_{i-1})$. Рассмотрим вершину v_i и все возможные дуги, входящие в вершину v_i. Обозначим множество начал таких дуг через $A^-(v_i)$. Если $A^-(v_i) = \emptyset$, т.е. нет дуг, входящих в вершину v_i, то нет (v_k, v_i)-путей и, следовательно, $OPT(v_i) = -\infty$. Пусть теперь $A^-(v_i) \neq \emptyset$. Что можно сказать о вершинах из множества $A^-(v_i)$? Номера таких вершин строго меньше i и для этих вершин мы уже нашли максимальные длины путей до них. Наидлиннейший (v_k, v_i)-путь проходит через одну из этих вершин. Рассмотрим наидлиннейший (v_k, v_i)-путь. Предпоследней вершиной в этом пути является некоторая вершина $v_t \in A^-(v_i)$. Вершина v_t делит путь на две части. Первая часть — от вершины v_k до вершины v_t, а вторая часть — от вершины v_t до вершины v_ℓ. Первая частья является наидлиннейшим (v_k, v_t)-путем и ее длина равна $OPT(v_t)$. Вторая часть состоит из одной дуги, длина которой $\ell(v_t, v_i)$. Общая длина (v_k, v_i)-пути равна $OPT(v_t) + \ell(v_t, v_i)$. Априори, мы не знаем какая именно вершина v_t из множества $A^-(v_i)$ является предпоследней в пути. Поэтому мы должны рассмотреть все возможные вершина $v_t \in A^-(v_i)$

и выбрать максимальную общую длину (v_k, v_t)-пути. Следовательно,

$$OPT(v_i) = \max_{v_t \in A^-(v_i)} \left(OPT(v_t) + \ell(v_t, v_i)\right), \quad \forall\, i \in \{k+1, k+2, \dots, \ell\}. \tag{2}$$

При этом будем запоминать вершину, на которой достигается максимум. Эта вершина является предпоследней вершиной в наидлиннейшем (v_k, v_i)-пути. Обозначим ее через $x(v_i)$. Таким образом, $x(v_i)$ — это вершина, на которой достигается максимум в (2). Используя рекуррентные соотношения (1) и (2), прямым ходом алгоритма динамического программирования найдем максимальную длину $OPT(v_i)$ для каждого $i \in \{k, k+1, \dots, \ell\}$. Значение $OPT(v_\ell)$ — это максимальная длина (v_k, v_ℓ)-пути. Если $OPT(v_\ell) = -\infty$, то в графе (v_k, v_ℓ)-путей нет. Пусть $OPT(v_\ell) \neq -\infty$. Как восстановить наидлиннейший (v_k, v_ℓ)-путь? Для этого достаточно использовать обратный ход алгоритма динамического программирования: v_ℓ, $x(v_\ell)$, $x(x(v_\ell))$, $x(x(x(v_\ell)))$, … , v_k.