Билет 17. Алгоритм Дейкстры поиска кратчайшего пути в графе и его сложность

Алгоритм Дейкстры — это алгоритм, который находит кратчайшие пути от одной из вершин графа до всех остальных. Алгоритм работает только для графов без ребер отрицательного веса $w(e) \neq 0$.

Пусть дан $G = (V, E)$ — ориентированный взвешанный связный граф, где V — множество вершин, E — множество ребер. Задача кратчайшего пути состоит в отыскании пути минимального веса, соединяющего начальную s и конечную t вершины графа G при условии, что хотя бы один такой путь существует.

На каждой итерации этого алгоритма любая вершина v графа G имеет метку $l(v)$, которая может быть постоянной либо временной. В первом случае $l(v)$ — вес кратчайшего (s, v)-пути. Если $l(v)$ временная метка, то $l(v)$ — вес кратчайшего (s, v)-пути, который прошел только через выршины с постоянными метками. Таким образом, временная метка $l(v)$ является оценкой сверху для веса кратчайшего (s, v)-пути, и став на некоторой итерации постоянной, она остается постоянной до конца работы алгоритма. $\theta(v)$ — номер вершины откуда пришел вес. После того как вершина t получила постоянную метку, с помощью метки $\theta(v)$ легко указать последовательность вершин, которые составляют кратчайший (s, t)-путь.

Перед началом итераций вершина s имеет постоянную метку $l(s)$=0, а все остальные вершины v - временную метку $l(v) = \infty$.

Общая итерация алгоритма: пусть p — вершина, получившая постоянную метку $l(p)$ на предыдущей итерации. Просматриваем все вершины $v \in \Gamma$(p), которые имеют временные метки, с целью уменьшить их(если это возможно). Метка $l(v)$ вершины $v \in \Gamma$(p) заменяется на

$$l(p) + w(p, v),$$

если

$$l(v) > l(p) + w(p, v).$$

В этом случаем говорим, что v получила метку $l(v)$ из вершины p, и пологаем $\theta(v) = p$.

Если же

$$l(v) \leq l(p) + w(p, v),$$

то метки $\theta(v)$ и $l(v)$ вершины v не меняются на данной итерации. Заканчиваем, когда метка $l(t)$ становится постоянной. $l(t)$ — вес кратчайшего (s,t)-пути, который обозначим как P^*. Этот путь определяется с помощью меток θ:

$$P^* = (s, ..., \theta_3(t), \theta_2(t), \theta(t), t),$$

где $\theta_k(t) = \theta(\theta(...\theta(v)...))$ $\forall v \in V$.

Алгоритм Дейкстры поиска кратчайшего пути

1. Положить $l(s) = 0$ — считать эту метку постоянной. Положить $l(v) = \infty$ для всех $v \in V$, $v \neq s$, эти метки временные. Положить $p = s$.

2. Для всех $v \in \Gamma$(p) с временными метками выполнить: если

$$l(v) > l(p) + w(p, v),$$

$$\text{то } (p) = l(p) = w(p, v) \text{ и } \theta(v) = p.$$

Иначе $l(v)$ и $\theta(v)$ не менять, где Γ(p) — множество соседних вершин к p, p - вершина, которая получила постоянную метку.

3. Пусть V' — множество вершин с временными метками l. Найти вершину v^*, такую что

$$l(v^*) = \min_{v \in V'} l(v).$$

Считать метку $l(v^*)$ — постоянной для вершины v^*.

4. $p = v^*$. Если $p = t$, то переходим к 5 пункту, где $l(t)$ — вес кратчайшего пути. Иначе переходим к пункту 2.

5.

$$P^* = (s, ..., \theta_3(t), \theta_2(t), \theta(t), t),$$

где P^* — кратчайший путь.

Теорема 1. *Алгоритм Дейкстры строит в графе G кратчайший (s,t)-путь.*

Доказательство. Пойдем по индукции.

1. При $k = 1$ — очевидно.

2. Пусть для $k-1$ верно. Обозначим через P^0 (s,v)-путь, построенный алгоритмом в результате k итераций, а через P^* — кратчайший (s,v)-путь. По условию $w(P^0) = l(v)$. Пусть V_1 и V_2 — множества вершин, у которых соответсвенно есть постоянные и временные метки перед началом k-й итерации. Рассмотрим два случая:

1—случай: Пусть P^* содержит вершину из V_2. Пусть $\bar{v}$ - первая вершина (считая от s), принадлежащая P^*, а вершина v' предшествует $\bar{v}$ на пути P^*, т.е. $(v', \bar{v}) \in EP^*$(т.е. ребро $(v', \bar{v})$ из множества ребер E на пути P^*). В силу выбора v' имеем $\bar{v} \in V_1$. Обозначим: P_1^* — часть пути P^* от s до $\bar{v}$. По предположению индукции $l(v')$ — вес кратчайшего (s,v')-пути. Поэтому

$$w(P_1^*) = l(v') + w(v', \bar{v}) \geq l(\bar{v}) \ (1),$$

т.к. $l(\bar{v})$ — временная метка, а $l(v)$ — постоянная метка вершины v выбирается на k-ой интерации как минимальная из временных, то

$$l(\bar{v}) \geq l(v) \ (2).$$

Объединим неравенства (1) и (2), также воспользуемся условием

$$w(P^0) = l(v),$$

подставим его в неравенство (1) вместо $l(\bar{v})$, получим

$$w(P_1^*) = l(v') + w(v', \bar{v}) \geq l(v) = w(P^0),$$

далее, из того, что P_1^* часть пути P^*, то очевидно, что

$$w(P^*) \geq w(P_1^*).$$

Отсюда следует:

$$w(P^*) \geq w(P_1^*) \geq l(v) = w(P^0),$$

т.е. P^0 — это кратчайщий (s,v)-путь.

2-случай: Все вершины P^* входят в V_1. Пусть v' и v'' вершины такие, что $(v', v) \in EP^*$, $(v'', v) \in EP^0$. Обозначим через P' — часть пути P^* от s до v', согласно предположению индукции имеем

$$w(P') \geq l(v').$$

Поэтому, если $v' = v''$, то

$$w(P^*) = w(P') + w(v', v) \geq l(\bar{v}) + w(v', v) = w(P^0).$$

Пусть теперь $v' \neq v''$. Т.к. v получает постоянную метку $l(v)$ из v'', а не из v' (т.к. алгоритм проходит через v'' к v, а не черезе v'), то

$$w(P^*) = l(v') + w(v', v) \geq l(v'') + w(v'', v) = w(P^0).$$

(Эта формула следует из того, что в вершину мы приходим из пути, который строит алгоритм, и минимального пути. А алгоритм хуже, чем минимальный путь не построит).
Таким образом, и в случае 2 верно неравенство

$$w(P^0) \leq w(P^*),$$

т.е. P^0 — кратчайший (s, v)-путь. Теорема корректности доказана. □

Теорема 2. *Алгоритм Дейкстры строит в графе G кратчайший (s, t)-путь за время $O(|G|^2)$*

(Напоминание: $|G|$ — мощность графа равная числу вершин n, т.е. $|G| = n$)

Доказательство. Вычисления максимальны, когда вершина t получает постоянную метку самой последней и G — является полным. Поэтому $|G| - 1$ - это число итераций алгоритма, т.е. каждый из пунктов 2 — 4 выполнятеся $|G| - 1$ раз. Очевидно, что пункт 4 выполеятся за время $O(1)$, а для выполнения каждого из пунктов 2,3 достаточно времени $O(|G|)$.

Построение пути с помощью меток θ можно сделать, затратив не более $O(|G|)$ операций. Отсюда следует, что время построения кратчайшего (s, t)-пути не превосходит $O(|G|^2)$. Теорема о сложности алгоритма доказана. □