Алгоритмы поиска кратчайших путей в графах

1. Стоимость пути во взвешенном графе, кратчайший путь.
   Свойства кратчайших путей и процедуры ослабления ребра:

   (a) Обозначения и определения:
   Путь: $p = v_1 \to v_2 \to \cdots \to v_n$
   Стоимость пути: $w(p) = \sum_{i=1}^{n-1} w(v_i, v_{i+1})$
   Стоимость кратчайшего пути из u в v: $\delta(u, v) = min\{w(p) : p$ — путь из u в $v\}$
   Если из u в v не существует пути, то $\delta(u, v) = \infty$
   Будем обозначать величиной $d[u]$ текущую оценку длины кратчайшего пути.
   "Родительский"граф G_p — подграф G, порождённый рёбрами $(p(v), v)$ для всех $v : p(v) \neq null$
   Деревом кратчайших путей (из вершины $s \in V$) графа $G = (V, E)$ называется каркасное дерево с корнем в s, такое что $\forall v \in V$ путь в этом дереве из корня s в вершину v является кратчайшим путём из s в v в графе G.

   (b) подпуть кратчайшего пути — тоже кратчайший путь
   Если $v_1, \dots, v_i, \dots, v_j, \dots, v_k$ — кратчайший путь (от v_1 до v_k), то $v_i, \dots, v_j$ — тоже кратчайший путь (от v_i до v_j)

   (c) неравенство треугольника
   $\delta(x, z) \leqslant \delta(x, y) + \delta(y, z) \quad \forall x, y, z \in V$

   (d) ограниченность снизу для оценки длины пути
   $d[v] \geqslant \delta(s, v) \quad \forall v \in V$

   (e) сходимость
   Пусть $s \to \cdots \to u \to v$ — кратчайший путь из s в v и u — вершина этого пути, предшествующая v.
   Тогда если в некоторый момент перед ослаблением ребра (u, v) было выполнено $d[u] = \delta(s, u)$, тогда после ослабления ребра (u, v) будет выполнено равенство: $d[v] = \delta(s, v)$

2. Задачу поиска кратчайших путей для невзвешеннных графов решает оычный обход графа в ширину. Время работы $O(V + E)$.

3. Алгоритм Дейкстры для ориентированных взвешенных графов с неотрицательными весами, $O(V \lg V + E)$.

   (a) Алгоритм является типичным примером т.н. жадных алгоритмов, т.е. делающих на каждом шаге наилучший локальный выбор.
   Пусть граф $G = (V, E)$ — ориентированный граф, и $w : E \to \mathbb{R}^+$ — неотрицательная функция.
   Пусть $S = \emptyset, s$ — заданная вершина, стоимость кратчайшего пути из которой требуется найти. Положим $d[s] = 0, d[u] = \infty$ для $u \neq s$

   (b) Пока $V \backslash \{S\} \neq \emptyset$:
   выбрать из $V \backslash \{S\}$ вершину u, с минимальным $d[u]$
   ослабить все рёбра, ведущие из u в вершины из $V \backslash \{S\}$
   добавить вершину u в S

   (c) Доказательство (нестрогое).
   Будем называть вершины u, для которых выполнено равенство $d[u] = \delta(s, u)$ — чёрными; нечёрные вершины u, для которых выполнено неравенство $d[u] < \infty$, т.е. такие, для которых хоть раз была выполнена процедура ослабления — серыми; а остальные вершины — белыми. Заметим, что во время выполнения алгоритма чёрные и белые вершины никогда не соединены ребром.
   Покажем по индукции, что сохраняется инвариант: для всех вершин $u \in S$ выполнено равенство: $d[u] = \delta(s, u)$.
   Перед началом цикла в S содержится единственная вершина s, для которой $d[s] = \delta(s, s) = 0$. Пусть на k-м шаге инвариант верен. На $(k + 1)$-м шаге мы выбираем вершину $u \in V \backslash \{S\}$ с минимальным значением $d[u]$ из всех вершин этого множества. Покажем, что это значение $d[u]$ совпадает с $\delta(s, u)$.
   Предположим, что это не так, и есть какой-то путь p короче, т.е. $w(p) < d[u]$. Поскольку первая вершина пути $s \in S$, а последняя $u \notin S$, то путь в графе, ведущий от s к u должен содержать ребро, соединяющее чёрную вершину (из S) с серой вершиной (не из S). Вершина u этой вершиной быть не может. Значит путь выходит из множества чёрных вершин через другую серую вершину (пусть это v), причём $d[v] \geqslant d[u]$. Тогда в силу минимальности выбора $d[u]$ должен существовать отрицательный путь из v в u, что невозможно, т.к. все веса неотрицательны.

   (d) Привести пример графа с отрицательным ребром, на котором алгоритм Дейкстры даст неверный результат.

4. Поиск кратчайших путей от заданной вершины в графе с отрицательными весами.

   (a) Алгоритм Беллмана-Форда. $|V| - 1$ раз выполняем ослабление всех рёбер графа. Порядок рёбер, в котором выполняется ослабление в течение одной итерации цикла, неважен (он может быть разным на каждой итерации). Алгоритм может выполняться параллельно (т.е. хорошо приспособлен для выполнения в распределённой системе).

   (b) Доказательство.
   Пусть $G = (V, E)$ — граф без отрицательных циклов. Тогда алгоритм Беллмана–Форда заканчивает работу и $d[u] = \delta(s, u), \ \forall u \in V$.

Достаточно показать, что для любой вершины $u \in V$ после какой-то из $|V|-1$ итераций будет выполнено равенство $d[u] = \delta(s, u)$ (дальше $d[u]$ уменьшаться не может, см. свойства процедуры ослабления).
Рассcмотрим $v \in V$ и кратчайший путь p из s в v : $p = v_0 \to v_1 \to v_2 \to \cdots \to v_k$ (здесь $s = v_0, v = v_k$). Если кратчайших путей между этими вершинами несколько, то выберем тот, который содержит наименьшее количество рёбер. Такой выбор пути гарантирует нам, что путь p — простой путь. В самом деле, предположим, что путь не простой. Отрицательных путей в графе (а значит в пути p) нет по условию. Если в пути p существует "положительный"цикл, то, выкинув его, мы получим путь короче чем кратчайший. А если в нём существует "нулевой"цикл, то, выкинув его, мы получим кратчайший путь, содержащий меньше рёбер. В обоих случаях получаем противоречие с условием выбора пути p.
Из оптимальности кратчайшего пути следует, что:

$$\delta(s, v_i) = \delta(s, v_{i-1}) + \omega(v_{i-1}, v_i) \tag{1}$$

Далее рассуждаем по индукции: $d[v_0] = 0, \delta(s, v_0) = 0$ (кратчайший путь не может быть отрицательным, т.к. тогда существует "отрицательный"цикл).
Предположим, что $d[v_j] = \delta(s, v_j)$ после j итераций внешнего цикла, где $j < i$. В частности, после $(i-1)$-й итерации выполнено $d[v_{i-1}] = \delta(s, v_{i-1})$. В течение i-й итерации мы ослабляем **все** рёбра, а значит и ребро $v_{i-1} \to v_i$. Но p - кратчайший путь, и ребро $v_{i-1} \to v_i$ в этом пути содержится. Из этого и 1 следует, что:

$$d[v_i] = d[v_{i-1}] + \omega(v_{i-1}, v_i) = \delta(s, v_{i-1}) + \omega(v_{i-1}, v_i) = \delta(s, v_i) \tag{2}$$

Таким образом, после k итераций мы гарантируем, что $d[v_k] = \delta(s, v_k)$. Но простой путь p в графе G не может содержать больше, чем $|V|-1$ ребро, следовательно $k \leqslant |V|-1$. А значит, после $|V|-1$ итераций цикла мы ослабим **все** рёбра во **всех** простых кратчайших путях графа G.

(c) Если отрицательный цикл отсутствует, то доказано, что $d[v] = \delta(s, v), \forall v \in V$. Если $\exists u \in V$, такая, что $d[u] > \delta(s, u)$, мы можем сделать вывод, что отрицательный цикл существует. Таким образом, если после работы алгоритма Беллмана–Форда попытаться ослабить каждое ребро графа (выполнить одну итерацию основного алгоритма) и какое-то ребро удастся ослабить, это означает, что в графе G есть отрицательный цикл.

(d) Оценка времени работы: $|V| * |E|$

5. Поиск кратчайших путей для ациклических ориентированных графов (топологическая сортировка + одна итерация Беллмана-Форда, $O(V + E)$)

6. Поиск кратчайших путей между всеми парами вершин. Алгоритм динамического программирования.

7. Поиск кратчайших путей между всеми парами вершин. Алгоритм Флойда-Уоршолла.
Пусть дан ориентированный граф $G = (V, E), |V| = n$, с весовой функцией $w_{ij} : E \to \mathbb{R}$. Требуется построить таблицу стоимостей кратчайших путей δ, где $\delta(i, j)$ — стоимость кратчайшего пути из вершины i в вершину j.
Обозначим d_{ij}^k — стоимость пути из вершины i в j, проходящего только через вершины из множества $\{1 \dots k\}$.
Заметим, что:

$$d_{ij}^0 = w_{ij}$$
$$d_{ij}^n = \delta(i, j)$$

Для решения применяется метод динамического программирования:

$$d_{ij}^k = min(d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1})$$