Алгоритм для задачи APSP с помощью схемной сложности.

5 мая 2019 г.

1 Abstract

APSP

Дано: граф G на n вершинах и весовая функция $w : [n] \times [n] \to \mathbb{R}_+ \cup \infty$

Вычислить: Для всех $i,j \in [n]$ кратчайший путь из i в j.

Определение 1 *Тропическое произведение матриц A и B - это матрица $A \odot B$ состоящая из элементов $(A \odot B)[i,j] = \min_k(A[i,k] + B[k,j])$.*

Теорема 1 *Для решения задачи APSP достаточно вычислить тропическое произведение матриц $A,B \in \mathbb{R}^{n\times n}$.*

Теорема 2 *$A \odot B$ вычисляется в классе AC_0, где AC_0 - это схемы константной глубины и полиномиального размера.*

Теорема 3 *(Разборов-Смоленский) Для любой AC_0-схемы C на n входах, размера s и глубины d, существует распределение $D(C)$ на полиномах степени $(log(s))^{O(d)}$ над $\mathbb{F}_2$, такое что $Pr_{p \sim D(S)}[p(x) = C(x)] \geq 3/4$*

Теорема 4 *Даны $A,B \subseteq \{0,1\}^m$, $|A| = |B| = n$ и полином $q(x_1, ..., x_m, y_1, ..., y_m)$ над $\mathbb{F}_2$, $|q| \leq n^{0.1}$, полином q можно вычислить на всех $(x,y) \in A \times B$ за $n^2 poly(log(n))$*

APSP алгоритм

Даны $A,B \in [m]^{n\times n}$, хотим посчитать $A \odot B[i,j] = \min_k(A[i,k] + B[k,j])$, пусть $d = 2^{(log(n))^\delta}$, $\delta > 0$

Шаг 1. По матрице A строим матрицы $A_1, ..., A_{n/d}$ размера $n \times d$, аналогично для B. Тогда $(A \odot B)[i,j] = \min_{k=1,...,n/d}(A_k \odot B_k)[i,j]$. При известных $(A_k \odot B_k)$ мы можем вычислить $(A \odot B)$ за $O(n^2 n/d)$ Далее мы хотим сделать d таким, что все тропические произведения $A_k \odot B_k$ будут вычисляться за $n^2 poly(log(n))$, тогда мы получим решение APSP за $O(n^3 poly(log(n)/d)$

Шаг 2. Даны A_k и B_k размера $n \times d$ и $d \times n$ соответсвенно. Пусть C - AC_0 схема для тропического произведения двух векторов длины d со значениями из $\{0, ..., n^k\}$ C сордержит $O(d * k * log(n))$ входов, $O(k * log(n))$ выходов и имеет размер $(d * k * log(n))^{O(1)}$.

Для любого выхода $j = 1, ..., O(k * log(n))$

a. Случайно выбираем $p_1^j, ..., p_{10log(n)}^j \sim D(C)$. Для любой пары i, j мы имеем $deg(p_i^j) \leq (log(d * k * log(n))$, тогда $|p_i^j| \leq (d * k * log(n))^{(log(d * log(n))^c}$

b. Оцениваем $p_1^j, ..., p_{10log(n)}^j$ на всех строках A_k и столбцах B_k.

c.Выводим majority бит.

Шаг **b.** можно сделать за $n^2 poly(log(n))$, при условии $(d * k * log(n))^{(log(d * log(n)))^C} \leq n^{0.1}$. Поэтому при $d = 2^{(log(n))^{1/1+c}/10}$ мы получаем APSP за $n^3 / 2^{\Omega(log(n))^{1/1+c}}$.