Лекция 6. Логарифмический алгоритм для UPATH

1 Общая идея

Первый шаг — доказать, что диаметр экспандера есть $O(\log_{\frac{1}{\lambda}})$ при константном λ.

Если степень экспандера константна, то все пути длины $O(\log N)$ можно перебрать за полиномиальное время на логарифмической памяти.

Следующая идея: с помощью зигзаг-произведения превращать граф в экспандер, сохраняя связность. В полученном экспандере проверим наличие пути перебором.

2 Диаметр экспандера

Утверждение 1. Если π — распределение верятностей, M — матрица случайного блуждания, $u = (\frac{1}{N}, \dots, \frac{1}{N})$, то $\|\pi M^l - u\|_2 \leqslant \lambda^l$.

Доказательство. $\pi = \pi^{\|} + \pi^{\perp} = u + \pi^{\perp}$. $\pi M^l = u + \pi^{\perp} M^l \Rightarrow \|\pi M^l - u\| = \|\pi^{\perp} M^l\| \leqslant \lambda^l \|\pi\| \leqslant \lambda^l \|\pi\|_1 = \lambda^l$. □

3 Приведение графа к экспандеру

Утверждение 2. Можно считать, что данный граф 3-регулярный.

Доказательство. Каждую точку, у которой меньше трёх соседей, дополним кратными петлями. Каждую точку, из которой выходит больше трёх ребер преобразуем в цикл длины равной её степени с торчащими рёбрами куда надо. □

Алгоритм будет следующий:

- Выберем граф H с параметрами $(D^4, D, \frac{3}{4})$, D — константа.
- D^2-регуляризуем граф, притом сделаем его не двудольным (петлей, например).
- $k = 1, \dots, l = O(\log N)$, $G_k = G_{k-1}^2 ⓩ H$, s_k, t_k — произвольные вершины из облаков s_{k-1}, t_{k-1}.
- Проверяем $s - t$ связность в экспандере перебором.

Утверждение 3. Алгоритм корректен.

Доказательство. Граф недвудольный и связный, значит λ отделено от 1, и после шага алгоритма все так и останется. Это рассуждение можно применить для каждой связной компоненты исходного графа, значит компоненты сохраняются.

Пусть C_k — компонента связности G_k, содержащая s_k. $\gamma(C_0) = \frac{1}{poly(N)}$. $\gamma(C_{k-1}^2) \geqslant 2\gamma(C_{k-1}) - \gamma^2(C_{k-1})$. Тогда:

$$\gamma(C_{k-1}^2 ⓩ H) \geqslant \frac{2 \cdot 9}{16}\gamma(C_{k-1})(1 - \frac{\gamma(C_{k-1})}{2}) \geqslant \min\left\{\frac{35}{32}\gamma(C_{k-1}), \frac{1}{18}\right\}$$

.

Для вычисления соседа в G_k нужен 1 переход в G_{k-1}^2 и 2 перехода в H. Переходы в H памяти не требуют, то есть в итоге получаем два перехода в G_{k-1}.

Логарифмическая память не зависит от модели вычислений, но доказать, что на каждой итерации добавляется константная память можно только в конкретной модели. Мы рассматрим такую модель: лента с исходным графом G, лента с u, i + дополнительная информация, рабочая лента. При запросе мы меняем (u, i) на (v, j), не меняя дополнительной информации. В такой модели нетрудно придумать, как вычислять квадрат и нормально так попотеть. По сути, утверждается, что рекурсия здесь почти хвостовая. □