

Удивительная алгебра сравнения строк (часть 2)

А. В. Тискин

DPhil (Oxford), доцент МКН СПбГУ

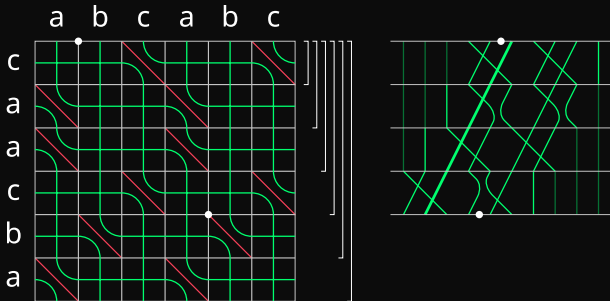
Б. Золотов

аспирант МКН СПбГУ

Dodrans-local LCS



Задача: для пары строк a, b предподсчитать оракул, который сможет быстро отвечать на запросы вида $\text{LCS}(a[0 : i_1], b[j_0 : j_1])$.



Посчитаем все префиксные \square -произведения перестановок, будем хранить в структуре данных, отвечающей на *range queries*.

Локальная задача LCS



Задача: для пары строк a, b предподсчитать оракул, который сможет быстро отвечать на запросы вида $\text{LCS}(a[i_0 : i_1], b[j_0 : j_1])$.

Запрос соответствует прямоугольнику на решётке, координаты вершин которого — $i_0, i_1; j_0, j_1$.

Local LCS: оракул Sakai

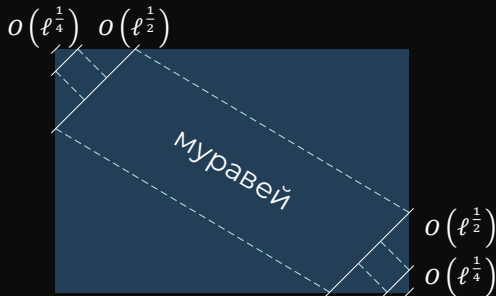


Предподсчёт за $\tilde{O}(n^2)$, запрос за $\tilde{O}(\sqrt{\ell})$.

Sakai, 2022

Здесь ℓ — размер прямоугольника, соотв. запросу.

Для уровней, делящихся на $\frac{n}{2^r}$, посчитаем перестановки между теми, разность которых не превосходит $\left(\frac{n}{2^r}\right)^2$.



Запрос — \square -перемножение подперестановок и *range queries*.

Local LCS: оракул Ch+



Charalampopoulos, Gawrychowski, Mozes, Weimann, 2021.

Если хранить перестановки в *дереве отрезков*, то между противоположными вершинами прямоугольника будет $\log n$ шагов по перестановкам.

Трудность — выбор индекса, в который приходит очередной шаг.

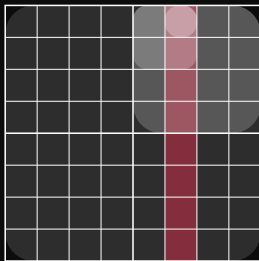
Ch+ используют *чёрный ящик MSSP*, а мы знаем, как улучшить время работы, применив муравья.

Динамическое выравнивание



Задача: обновлять LCS при вставке/удалении символа произвольной из двух строк

Иерархия, в которой у каждого прямоугольника четыре потомка. Внутри каждого — посчитана перестановка.



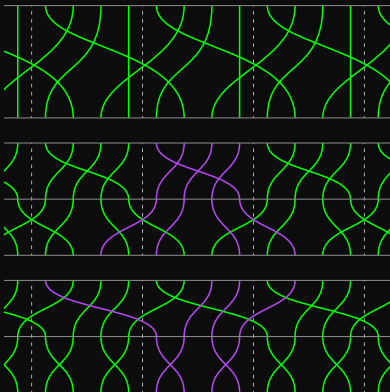
Суммарный размер перестановок, изменённых при вставке/удалении символа, на каждом уровне иерархии, — $O(n)$. *Charalampopoulos, Kociumaka, Mozes, 2020*

Аффинные перестановки



Научимся работать с перестановками в аффинном моноиде Гекке.

Gaevoy, Tiskin, Zolotov, 2025

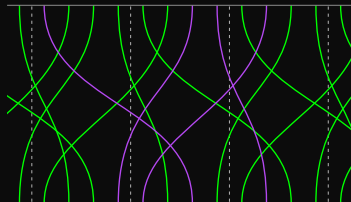
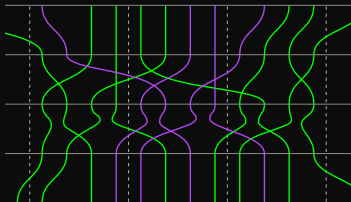
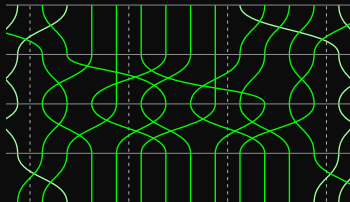
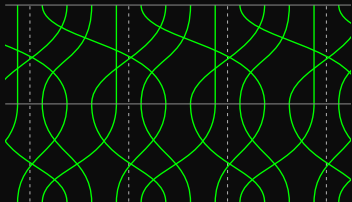


Аффинная перестановка раскладывается на произведение *перестановки конечного типа* и *грассмановой перестановки*.

Аффинное \square -умножение



Можно свести \square -умножение аффинных перестановок к обычному \square их трёх соседних периодов.



Периодическая задача LCS



Задача: найти $\text{LCS}(a^k, b^m)$.

- Возвести перестановку в \square -степень k ;
- посчитать, сколько копий каждой нити пересекает m периодов.

Время — $O(|a| \cdot |b| + |b| \cdot \log |b| \cdot \log k)$.

Приближенный поиск подстроки



Charalampopoulos, Kociumaka, Wellnitz 2022

Задача: найти в тексте T подстроки, отличающиеся от шаблона P не более чем на редакционное расстояние k .

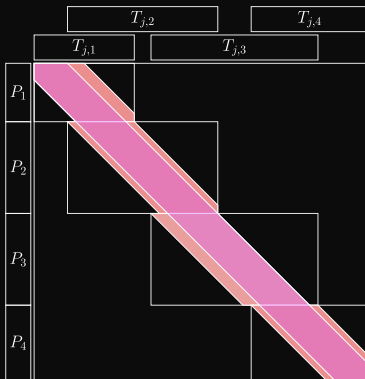
Ключевой шаг решения этой задачи — *dynamic puzzle matching* строк с малым редакционным расстоянием:

Дана эталонная строка U и семейство \mathcal{F} . Известно, что $\sum_{u \in \mathcal{F}} \delta(u, U) = O(k)$. Последовательность пар $(P_1, T_1) \dots (P_Z, T_Z)$; $P_i, T_i \in \mathcal{F}$; пары могут в неё динамически вставляться и удаляться. Поддерживать вхождения $P_1 \dots P_Z$ в $T_1 \dots T_Z$ с не более чем k редакциями, быстро обновлять при вставке/удалении пары.

Dynamic puzzle matching



Не отходим более чем на k от главной диагонали —
монжевы матрицы расстояний размером $O(k)$.



Построение: малое суммарное δ — применим
динамическое выравнивание, чтобы построить матрицы
расстояний для всех возможных пар.

Спасибо за внимание!



- ***Dodrans-local LCS***: храним много перестановок, полученных муравьём
- ***Local LCS***: дерево отрезков из перестановок с операцией \square
- ***Dynamic LCS***: дерево отрезков из перестановок переменного размера
- ***Periodic LCS***: аффинное \square -умножение и его непосредственное применение
- ***Approximate pattern matching***: применение динамического выравнивания при ограниченном редакционном расстоянии

https://t.me/boris_a_z