Суффиксный автомат

Алгоритм

- Определение, несколько примеров вручную построенных автоматов
- Худшие тесты по количеству состояний и по количеству переходов
- Дополнительные параметры: длина состояния length и суффиксная ссылка link
- Описание структуры данных для хранения автомата
- Построение автомата для пустой строки
- Алгоритм sa_extend перестроения автомата при добавлении символа в конец строки: три принципиальных случая с неформальными доказательствами; реализация алгоритма
- Доказательство фактов, применявшихся в разработке алгоритма

Применения

- Быстрый поиск подстроки в тексте: за O от длины подстроки при условии выполненного препроцессинга
- Та же задача, но когда автомат построен для подстроки, и по нему прогоняется текст
- Поиск наименьшего циклического сдвига (за O(n) — с выводом его)
- Подсчёт количества различных подстрок (простая линейная динамика)
- Подсчёт суммы длин всех различных подстрок (две похожие линейные динамики)
- Поиск кратчайшей строки, не входящей в данную в качестве подстроки (впрочем, эту задачу можно решить и более прямолинейно; здесь же — с помощью простой линейной динамики)
- Внесение в автомат доп. информации для поиска позиции первого вхождения подстроки (надо сделать pos — позиции окончаний строк состояния)
- Внесение в автомат доп. информации для поиска количества вхождений подстрок (проставить динамику для всех неклонированных состояний в 1, для остальных в 0, а затем произвести p->link->cnt += p->cnt для всех состояний p; либо сделать граф инвертированных суффиксных ссылок и просто считать динамику рекурсивно по нему)
- Внесение в автомат доп. информации для поиска всех вхождений подстроки за время O от их количества (надо взять те же posы, и ещё граф инвертированных суфф. ссылок; тогда пустить dfs по нему из текущего состояния, и все posы достигнутых неклонированных состояний будут окончаниями вхождений, причём без повторений)