1 Условие

Дано ориентированное (подвешенное) дерево из n вершин, на каждом ребре написан символ. Найти число путей, идущих вниз по дереву, соответствующих строке s. Алфавит константного размера, время $\mathcal{O}(n + |s|)$.

2 Решение

Заметим, что каждый путь от корня до листа задает некоторую строку. Обозначим множество всех таких строк A. Тогда задача сводится к нахождению количества вхождений s во все строки $t \in A$.

Рассмотрим быстрый ($\mathcal{O}(|s|)$) алгоритм построения префикс-функции.

```
p[0] = 0
for i = 1 to s.length - 1
    k = p[i - 1]
    while k > 0 and s[i] != s[k]
        k = p[k - 1]
    if s[i] == s[k]
        k++
    p[i] = k
```

Мы можем применить этот алгоритм ко всему дереву, т.к. алгоритм либо поднимается вверх (`k = p[k - 1]`), либо спускается на единицу (`k++`). Спуск вниз можно реализовать наподобие dfs — спускаемся во всех детей по очереди. Но подъем вверх *(на произвольную высоту)* необходимо делать за константу, что нетривиально.

Задача подъема в дереве — Level ancestor problem и она решается в предподсчёт $\mathcal{O}(n)$, запрос $\mathcal{O}(1)$. Общая идея состоит в том, чтобы разложить дерево по четырём русским, потом разложить его в непересекающиеся длинные пути, потом увеличить длину путей вверх в два раза и двоичными подъемами найти ответ для каждого пути.

Таким образом, мы можем посчитать префикс-функцию для всего дерева. Итоговый алгоритм следующий:

1. Подвесим корень дерева к пути, построенному из строки $s\#$
2. Посчитаем префикс-функцию для дерева
3. Обойдём дерево и посчитаем количество вершин v, таких что $p[v] = n$