

Задача 4.

**Чемпионат по поиску в
сети Меганет**

По набору **фильтров** и списку **адресов** найти для каждого адреса число фильтров, которым он соответствует.

Адрес имеет вид: **имя-сервера имя-раздела** (слитно, без пробела).

имя-сервера содержит от 1 до 5 разделенных точкой непустых частей из строчных латинских букв, например: **a, ab.cd, abacaba, a.b.c.d**

имя-раздела либо пусто, либо содержит от 1 до 5 частей из строчных латинских букв, начинающихся символом **/**, например: **, /a, /aba, /a/b/c/d/e**

Примеры адреса:

a aba/d/f/g/h a.b aba.caba/def/g c.d.e.f.g/a/b/c/d/e

Фильтр имеет вид:

фильтр-сервера фильтр-раздела (без пробела)

фильтр-сервера имеет вид: **S** или ***.S** ,
где **S** - *имя-сервера*.

Фильтру сервера вида ***.S** соответствуют сервера, удалением нуля или более начальных частей имени которых можно получить строку **S**.

Аналогично **фильтр-раздела** имеет вид:

R или **R/*** ,

где **R** - *имя-раздела*, а **/*** - соответствует нулю или более конечных частей имени раздела.

Адрес соответствует фильтру, если его имя сервера соответствует фильтру сервера, а его имя раздела соответствует фильтру раздела.

Таким образом, префиксу (началу строки) ***.** фильтра сервера соответствует в адресе любая (в том числе пустая) последовательность начальных частей имени сервера. Суффиксу (концу строки) **/*** фильтра раздела соответствует в адресе любая (в том числе пустая) последовательность конечных частей имени *раздела*.

Фильтр	Примеры соответствующих адресов		
ab.c/d/e	ab.c/d/e		
*.a	a	ax.a	efg.a
*.a/b/c	a/b/c	x.a/b/c	efg.a/b/c
x.yz/a/*	x.yz/a	x.yz/a/b/c	x.yz/a/xyz
.a/	a	x.a	e.fg.a
	a/b/c	x.a/ddd/c	e.fg.a/b/c/g/haha/l
.a/b/c/	a/b/c	x.a/b/c	e.fg.a/b/c
	a/b/c/xxx	e.fg.a/b/c/d/e/f	

Примеры входных и выходных файлов

search.in search.out

2 0

a.bb/c 1

bb/c/d 0

4 0

a.bb

bb/c/d

a.bb/c/d

bb/c

search.in search.out

4 0

*.bb/c 4

.bb/c/ 3

bb/c/* 0

bb/c/* 2

6 1

bb

bb/c

bb/c/d

a.bb

a.bb/c

a.bb/c/d

Идеи решения

Имя сервера и **имя раздела** состоят не более чем из 5 частей каждое, и адрес подходит под фильтры, где **фильтр сервера** совпадает с **именем сервера** или содержит некоторое количество его заключительных частей, перед которыми идет звездочка - не более 6 вариантов (без звездочки, оставить 1, 2, ..., 5 конечных частей).

Фильтр раздела может либо совпадать с **именем раздела**, либо содержать некоторое количество его начальных частей, после которых идет звездочка, - не более 7 вариантов (без звездочки, оставить 0, 1, 2, ..., 5 начальных частей).

Поэтому для каждого адреса может подходить **не более $6 \cdot 7 = 42$** различных фильтров.

Частичные решения основаны на поиске подстроки в строке, переборе всех вариантов, хешировании. Разместив фильтры, например в **боре** или в структуре C++ **`std::set`**, можно за $O(1)$ или $O(\log n)$ проверять наличие фильтра. Следует учитывать, что может быть несколько одинаковых фильтров.

Для подзадачи 1 достаточно любым методом проверить каждый фильтр после удаления концевых звездочек - является ли он подстрокой адреса.

Максимально эффективные решения должны либо **быстро проверять вхождения подстроки в строку**, например, алгоритмом Кнута-Морриса-Пратта, либо **учитывать, что «стык» имени сервера и имени раздела находится однозначно и можно проверять лишь вхождения, где он правильно расположен.**

В тестах к **подзадаче 2** для каждого адреса может подходить только один фильтр, полностью с ним совпадающий. Для решения можно применить **любую стандартную структуру данных**, например, поместить все фильтры в структуру C++ map <string, int> и искать число подходящих фильтров одним запросом к этой структуре.

Для решения **подзадачи 3** необходимо применить подход, описанный в **эффективном решении** выше. Ограничения по времени в этой задаче достаточно жесткие, различные неасимптотические неточности в реализации или неудачное использование хеширования для сравнения строк приводят к тому, что часть тестов может не пройти, такие решения получают лишь частичную оценку.

Литература

1. *Кормен Т., Лейзерсон Ч., Ривест Р.* Алгоритмы: построение и анализ. - М.: МЦНМО, 2000. - 960 с.
2. *Окулов С. М.* Алгоритмы обработки строк: учебное пособие. - М.: БИНОМ. Лаборатория Базовых Знаний, 2009.
3. *Гасфилд Д.* Строки, деревья и последовательности в алгоритмах: Информатика и вычислительная биология / Пер. с англ. И. В. Романовского. - СПб.: Невский диалект; БХВ-Петербург, 2003.
4. *Смит Б.* Методы и алгоритмы вычисления на строках. – М.: ООО «И. Д. Вильямс», 2006.

Спасибо за внимание