Стринг (анализ)

Нека искаме да изчислим срещанията на C в елемента Z и знаем колко пъти се среща стрингът C в елементите Y и X, които са съответно предният и по-предният елемент. Да казваме, че F(s) е броят срещания на C в стринга s. Трябва да намерим F(Z). Нека също броят на X-овете в R е равен на Cx и броят на Y-ците в R – на Cy.

Когато елементите са достатъчно малки (до 1000000 символа), може лесно да се намери колко пъти се среща C в тях с алгоритъм като KMP(алгоритмът на Knuth-Morris-Pratt). Иначе, срещанията трябва да се намират по друг, по-умен начин.

Очевидно, F(Z) = Cx*F(X) + Cy*F(Y) + [новите срещания, които се получават между X и Y]. Например:

```
C = meow
X = eowm (F(X) = 0)
Y = eowmeowasm (F(Y) = 1)
R = XYX (Cx=2, Cy=1)
Z = eowmeowmeowasmeowm (F(Z) = 2*0 + 1*1 + 2 = 3)
```

Тъй като е вярно, че винаги |Z| ≥ |Y| и |Y| ≥ |X| (т.е. дължината им расте монотонно), можем да разгледаме 4 отделни случая:

Случай 1: |X| ≤ |Y| ≤ |Z| ≤ |C|

Тогава C не може да се среща в Z и очевидно F(Z) = 0. Само намираме стойността на Z и продължаваме. Сложността на това е O(|C|). Тъй като |Z| расте експоненциално (ако |R|=2, което е минималната стойност, |Z| е равно на числата на Фибоначи; представете си за |R|=100), тази стъпка няма да се повтори повече от 30 пъти.

Случай 2: |X| ≤ |Y| ≤ |C| ≤ |Z|

Абсолютно винаги |Z| ≤ Cx*|X| + Cy*|Y|. Тук |Z| не е ограничено, а |X|, |Y| ≤ |C|, затова |Z| ≤ Cx*|C| + Cy*|C| = |R|*|C|. Така в най-лошия случай |Z| ≤ |R|*|C|. Тъй като |R| ≤ 100 по условие, за да се запише цялото Z може да не стигне паметта, защото |R|*|C| ≤ 100*1000000 = 100 000 000 ~ 100MB, а ограничението за памет е 32MB. За щастие, това не е и нужно. Тъй като KMP работи с крайни автомати, можем просто да му даваме поредните букви в Z без да помним цялото Z и KMP ще си свърши работата. Това става като минаваме през R и когато срещнем ‘X’, минаваме през X, а когато срещнем ‘Y’, минаваме през Y. Сложността е O(|Z| + |C|) ~ O(|R|*|C| + |C|) ~ O(|R|*|C|).

Нека P(s) е първите |C|-1 символа на низа s, а S(s) е последните му |C|-1 символа. Тогава трябва да се запомнят P(Z) и S(Z), защото ще са нужни по-късно.

Случай 3: |X| ≤ |C| ≤ |Y| ≤ |Z|

По очевидни причини, вторият случай се среща само веднъж и след него идва този. Този случай е най-трудният (поне според мен). Трябва да се сметнат F(Z), P(Z) и S(Z) като се знае какви са X, F(X), P(Y), S(Y) и F(Y).

Тъй като този случай е сравнително сложен, ще използваме пример:

R = “XYYXXYXXXXYXYXXX”

Z = XYYXXYXXXXYXYXXX (не се записва в паметта, разбира се)

Тогава:

F(Z) = F(XP(Y)) + F(Y)
+ F(S(X)P(Y)) + F(Y)
+ F(S(Y)XXP(Y)) + F(Y)
+ F(S(Y)XXXXP(Y)) + F(Y)
+ F(S(Y)XP(Y)) + F(Y)
+ F(S(Y)XXX)

Тъй като S(s) и P(s) имат дължина най-много |C|-1, то F(S(s1)P(s2)) ще е броят на срещанията на C, които включват части както от s1, така и от s2, защото |C|-1 < |C|. Поради това F(S(Y)XX...XXP(Y)) не включва срещанията на C в самите копия на Y в Z, но включва всичките срещания на C между двата Y-ка и X-овете.

Както в предната стъпка, ще запомним P(Z) = P(XP(Y)) и S(Z) = S(S(Y)XXX).

Тази стъпка отново има сложност O(|R|*|C|).

Случай 4: |C| ≤ |X| ≤ |Y| ≤ |Z|

Случай 3 също се среща само веднъж. След него идва случай 4, който продължава до безкрайност. Сега се знаят F(X), F(Y), P(X), P(Y), S(X) и S(Y) и трябва да се намерят F(Z), P(Z) и S(Z).

Най-важното наблюдение е че P(Z) ще бъде равно или на P(X), или на P(Y), в зависимост от това дали R започва с ‘X’, или с ‘Y’. Аналогично, S(Z) = S(X) или S(Z) = S(Y). В такъв случай няма нужда да се презаписват всеки път суфиксите и префиксите, ами може просто да се запишат в един масив P(X), P(Y), S(X) и S(Y) и да се помнят само индекси, „сочещи“ към тях.

С малко мислене може да се заключи, че

F(Z) = Cx*F(X) + Cy*F(Y)
+ Cxx*F(S(X)P(X))
+ Cxy*F(S(X)P(Y))
+ Cyx*F(S(Y)P(X))
+ Cyy*F(S(Y)P(Y))

, където Cxx е броят на срещанията на “XX” в R и аналогично за Cxy, Cyx и Cyy.

Ако смятаме F(S(X)P(X)), F(S(X)P(Y)), F(S(Y)P(X)) и F(S(Y)P(Y)) всеки път, сложността на една стъпка в този случай ще е O(|C|). Тъй като според предното наблюдение те не се променят, обаче, можем да ги сметнем още когато навлезем в случай 4 за първи път и вече за O(1) операции да смятаме F(Z).

Така общата сложност на това решение е O(30|C| + 1*|R|*|C| + 1*|R|*|C| + K*1) ~ O(|R|*|C| + K).*