



Назив проблема: Символи

Аутор: Петар Величковић
Анализа: Петар Величковић
Тагови: кумулативне суме, sweep

Решење и анализа. Задатак Символи представља проблем средње тежине на овогодишњим Квалификацијама. На њему је било релативно једноставно освојити значајну количину бодова, док је за пуне бодове било потребно применити врсту идеје која није моментално интуитивна, али се често “провлачи” у такмичарским задацима, и зато искуснији такмичари углавном нису имали проблема са овим задатком. Аутор је оригинално смислио овај задатак 2013. године, инспирисан задацима “Мењање стринга” и “Мењање стринга v2” који су се појавили на тадашњим Квалификацијама, и представљају сличне примере предности паметних оптимизација над директном симулацијом.

Решење које се одмах намеће, с обзиром на мали број симбола (26), је да се пронађе решење за сваки могући Перичин одабир симбола, узимајући максимално од тих решења као коначно решење. Самим тим, наше решење се онда своди на решавање потпроблема у коме је унапред познато који симбол је одабрао Перица.

Директна симулација процеса у тексту задатка (тј. директно обртање сваког могућег стринга дужине k) може донети било који број бодова у интервалу од 20 до 60, у зависности од нивоа “оптимизованости” решења; ево неколико таквих решења:

- Решење које за сваки подстринг дужине k обрће све симболе у њему, а затим броји појављивања симбола додатним пролазом кроз цео низ; ово решење има укупну сложеност $O(n^2 \cdot k)$.
- Модификовано претходно решење у коме немамо додатни пролаз кроз цео низ на крају; искоришћавамо чињеницу да, уколико претходно запамтимо колико има ког симбола у целом низу, треба директно ажурирати ове суме само над интервалом који тренутно обрћемо да бисмо пребројали сва појављивања симбола. Ово решење има укупну сложеност $O(n \cdot k)$.
- Можемо додатно оптимизовати претходно решење уколико искористимо чињеницу да ће количине сваког симбола увек бити исте за исто обртање (тј. раније смо 26 пута обртали стринг на истом месту—и сваки пут добијали исте количине симбола). Уколико прво одрадимо обртање на некој позицији па тек онда израчунамо релевантне количине за сваки симбол на тој позицији, и даље имамо алгоритам сложености $O(n \cdot k)$, али ће овај алгоритам свеукупно извршити знатно мање операција обртања!

За преосталих 40 поена неопходно је искочити из оквира директне симулације задатка, и смислити оптималније решење. Једна од могућих идеја је да, уколико за сваки симбол памтимо *кумулативну суму* његових појављивања кроз цео низ, тј. за сваки симбол s и позицију $1 \leq x \leq n$, памтимо вредност K_x^s , тако да важи:

$$K_x^s = \sum_{i=1}^x \mathbb{I}(S_i = s)$$

где је S_i i -ти симбол у почетном стрингу, а $\mathbb{I}(S_i = s)$ је дефинисан као

$$\mathbb{I}(S_i = s) = \begin{cases} 1 & S_i = s \\ 0 & S_i \neq s \end{cases}$$

За сваки симбол s , ове вредности се могу израчунати у једном пролазу кроз низ; за сваки индекс, узимамо претходну вредност, и додајемо 1 уколико је тренутни симбол једнак симболу s , тј.

$$K_{x+1}^s = K_x^s + \mathbb{I}(S_{x+1} = s).$$



Када имамо ове вредности, могуће је ефикасније израчунати ефекат једног обртања без да се заправо директно обрну вредности стринга. Наиме, укупан број појављивања симбола s између позиција a и b је:

$$\sum_{i=a}^b \mathbb{I}(S_i = s) = \sum_{i=1}^b \mathbb{I}(S_i = s) - \sum_{i=1}^{a-1} \mathbb{I}(S_i = s) = K_b^s - K_{a-1}^s$$

Такође, укупан број појављивања симбола s је дат као:

$$\sum_{i=1}^n \mathbb{I}(S_i = s) = K_n^s$$

Означимо сада обрнути симбол симболу s са \bar{s} . Предност за обртање на i -тој позицији се онда може израчунати као:

$$\begin{aligned} & (K_n^s - (K_{i+k-1}^s - K_{i-1}^s) + (K_{i+k-1}^{\bar{s}} - K_{i-1}^{\bar{s}})) - (K_n^{\bar{s}} - (K_{i+k-1}^{\bar{s}} - K_{i-1}^{\bar{s}}) + (K_{i+k-1}^s - K_{i-1}^s)) \\ & = K_n^s - 2(K_{i+k-1}^s - K_{i-1}^s) + 2(K_{i+k-1}^{\bar{s}} - K_{i-1}^{\bar{s}}) - K_n^{\bar{s}} \end{aligned}$$

Пошто смо већ израчунали све ове вредности, за фиксирано обртање и Перичин симбол можемо одредити решење у константном времену. Ово нам свеукупно даје алгоритам оптималне временске сложености $O(n)$.

Међутим, наивно чување свих вредности K_n^s доводи до пробијања меморијског ограничења; уколико такмичари користе статички алоциране низове, онда ће моћи ово одмах да виде и коригују. Уколико корекција буде коришћење мање широког типа (нпр. `short/integer` уместо `int/longint`) ово ће довести до прекорачења опсега на већим примерима и до нетачног решења; оваква решења онда доносе око 80 бодова.

За оптимално решење, можемо приметити да, када разматрамо један симбол s , занимају нас само вредности K_n^s и $K_n^{\bar{s}}$, тј. само два симбола. Уколико у сваком моменту чувамо ове вредности само за ове симболе, смањујемо додатну меморијску сложеност алгоритма са $O(26n)$ на $O(2n)$, што је довољно за освајање 100 поена.

Напоменимо на крају да је могуће додатно модификовати оптимално решење, тако да је неопходно само *константно много* (тј. $O(1)$) додатне меморије. Да ли можете смислити ово решење?