`divisors0` и `divisors1`

Най-лесното решение е за всеки префикс да питаме за всички негови делители. От страна на имплементация обаче е по-лесно за всеки потенциален период да питаме за всички префикси с дължина кратна на периода. Това решение изкарва само 6 точки, но с една лесна оптимизация стига до 13 точки: разглеждаме възможните периоди във възходящ ред и не питаме за префикси с вече открити периоди.

`divisors2`

Нека дефинираме „частичен“ период по същия начин като период, но без изискването за делимост. Лесно се вижда (от дефиницията), че ако позиция L има частичен период P, то и всички позиции преди L също го имат. Следва, че ако позиция L няма частичен период P, то и всички позиции след L също го нямат. Също така забележете, че ако позиция има период P, то тя има и частичен период P.

От горните наблюдения можем да видим, че ако позиция kP няма период P, то и позиция $(k+1)P$ също няма да има период P. Следва, че можем да разглеждаме позициите $2P, 3P, \ldots$ във възходящ ред и, когато стигнем до kP, което няма период P, можем да прекъснем търсенето. Това решение ни носи 47 точки.

`divisors3`

Нужното наблюдение тук е, че ако позиция P има период $Q < P$, то P не е минималният период на никоя следваща позиция, т.е. можем да не правим търсене за период P. Това е защото, ако следваща позиция kP има период P, тя със сигурност ще има и период Q. Това решение получава 69 точки.

`divisors4`

Нека позиция $kP > P$ има минимален период P. Следва, че всички позиции до kP имат частичен период P. Сега, нека разгледаме позиция tQ между $Q+P$ и kP, за която твърдим, че има минимален период $Q \neq P$. Тя обаче има и частичен период P. Лесно се вижда, че от това следва следното: tQ има период $\gcd(Q, P)$ (може да се опитате да си разпишете контрапример). Ако Q не дели P, то $\gcd(Q, P) < Q$, която е противоречие, защото приехме, че Q е минималният период на tQ. Следва, че Q трябва да дели P. От това обаче следва, че позицията P трябва също да има период Q (тъй като $P < tQ$ и tQ има частичен период Q, P също има частичен период Q). Но това води до противоречие по съображенията направени в предното решение. Заключваме, че никоя позиция между $2P$ и kP няма минимален нетривиален период $Q \neq P$.

Сега нека да приемем, че позиция $sQ > Q$ има период Q, такъв че $P < Q < (k-1)P$. Разглеждаме следните варианти. Първо, ако $2Q < kP$, стигаме до противоречие от горния тип на позиция $2Q$, защото $2Q > P + Q$. Второ, ако $kP \leq 2Q$, стигаме до противоречие на позиция kP (с ролите на Q и P разменени), защото $kP = (k-1)P + P > Q + P$. Така заключваме, че винаги се стига до противоречие, т.е. никоя позиция не може да има период Q между P и $(k-1)P$. (Забележете, че случаят $Q = (k-1)P$ бива изключен още от съображенията направени в предното решение.)

Стигаме до следното решение: правим това, което прави и `divisors3`, но когато прекъснем търсенето за период P с финална открита позиция kP, следващият период, който разглеждаме, не е $P+1$, ами е $(k-1)P+1$. Това решение изкарва 84 точки.

Анализ на `divisors4`

Нека сега анализираме броя заявки, които това решение извършва. В началото си „заделяме“ $N/2$ заявки, по една за всеки възможен период. Когато разглеждаме период P, който е минимален на позиция kP, използваме общо k заявки, но пък елиминираме всички периоди до $(k-1)P$, както и kP (поради съображенията направени в предното решение). При $k=1$, единственият ново-елиминиран период е P, но това е в „бюджета“ ни, защото за него има заделена една заявка. При $k \geq 2$, общата бройка ново-елиминирани пероди е $(k-1)P+1-(P-1)=(k-2)P+2 \geq k$, т.е. отново се придържаме към бюджета си. Всичко това ОБАЧЕ се случва само докато текущата позиция е преди $N/2$, защото иначе няма периоди за елиминиране. Ако няма нетривиални периоди след позиция $N/2$, това не е проблем и се побираме в заделените си $N/2$ заявки.

Сега нека разгледаме случая да има нетривиален период P след позиция $N/2$. Това значи, че не може да има други нетривиални периоди след него. В най-лошия случай са ни трябвали $P+N/P-2$ заявки за целия масив. Първите $P-1$ се използват преди достигане на периода P (най-лошият случай е всички периоди преди P да са единици), а следващите $N/P-1$ – след това (най лошият-случай е да питаме за всяка кратна на P позиция след P).

Лесно се вижда, че тази функция на P е изпъкнала, т.е. максимумът ѝ в някакъв интервал е равен на стойността ѝ в един от двата края на интервала. Ясно е, че $P \leq N/2$. Следва, че това решение прави не повече от $N/2$ заявки, ако приемем, че $P \geq 2$. За жалост при $P=1$, то може да достигне $N-1$ заявки. Т.е. лошият му случай е, когато над половината от масива има период равен на едно.

`divisors5`

Горният анализ ни показва, че предното решение се „чупи“ в само един много конкретен клас от случаи. Най-лесното решение на този проблем веднага трябва да ни е очевидно: като първа заявка ще проверим дали сме в лош случай, т.е. ще проверим дали позиция $N/2$ има период равен на едно. Ако не, то не сме в лош случай и можем да процедираме както преди. Ако да, то знаем, че всички позиции до $N/2$ имат единичен период и остава линейно да проверим за всички следващи (общо $N/2$ заявки за тази дясна част). И в двата случая правим не повече от $N/2+1$ заявки. Това е пълното решение.

Динамично програмиране

Едно частично решение, предложено от Радослав Димитров, се базира на идея напомняща на динамично програмиране. Ако позиция $kP > P$ има минимален период P, то и позиция $(k-1)P$ трябва да има минимален период P. Така можем да инициализираме периодите на всички позиции да са равни на самите им индекси. След това, когато достигнем до позиция L (по индукция) вече знаем минималният ѝ период P. Остава да проверим дали това е период и на позиция $L+P$ и, ако да, ъпдейтваме периода ѝ. Забележете, че преди това ъпдейтване (преди проверката дори) текущият минимален период на $L+P$ винаги ще е равен на $L+P$, т.е. никога няма да ъпдейтваме дадена позиция над веднъж. Това позволява ъпдейта да е просто присвояване на стойността на P, а не ъпдейтване като минимума на P и предишната стойност. Това е следствие на наблюденията ни за `divisors4`, но тук не се възползваме от тях и затова това решение не е толкова добро. То изкарва 69 точки, а с оптимизация подобна на тази за divisors5 – 83 точки.

Долна граница

Вече разгледахме как да решим задачата, но дали това наистина е най-доброто възможно решение? Можем ли да покажем, че не е възможно да съществува по-добро решение? Ами, почти! Можем лесно да покажем долна граница от $N/2$ заявки. Да разгледаме следния случай: има не повече от една позиция с нетривиален период. Очевидно тази позиция трябва да е четна (периодът се повтаря точно два пъти), но не можем да заключим нищо повече за нея. Следва, че не можем да я намерим по по-ефикасен начин от това да питаме за всички четни позиции $2P$ дали имат период P. Това отнема $N/2$ заявки за тази опростена версия на пълната задача, т.е. и пълната задача не може да се реши в под $N/2$ заявки. Разумно е да допуснем, че по-сложната пълна задача изисква поне още една допълнителна заявка (общо $N/2 + 1$), в който случай нашето решение би било оптимално, но нямаме доказателство за това. Във всички случаи то е много близо до оптимално.

Автор: Емил Инджев
Начална идея: Радослав Димитров